SOFTALK

Chairman John Haller
Editor Margot Comstock Tommervik
Managing Editor Al Tommervik
Associate Editor Craig Stinson
Art Director Kurt A. Wahlner
Contributing Editors Roger Wagner
Jim Merritt
Frank Malone
Robert Koehler
William Depew
Guest Reviewer Joe Villareal
Assistant Editor Robert Koehler
Circulation David Hunter
Ron Rennells
Systems William V.R. Smith
Advertising Sales Al Tommervik
213-980-5074 Margot Tommervik

Cover photo of the power yacht *Westward* by Jack West.

Composition by Photographics, Hollywood, California. Printing by California Offset Printers, Glendale, California.

Apple is a registered trademark of Apple Computer Inc., Cupertino, California.

UCSD Pascal is a trademark of the University of California at San Diego.

VisiCalc is a trademark of Personal Software, Sunnyvale, California.

Subscriptions: Complimentary to all owners of Apple computers in the USA. Apple owners in Canada and Mexico: $8 per year; other Apple owners outside the USA: $15 per year, first-class airmail.

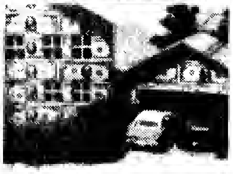
Back Issues: $2 through February 1981; $2.50 thereafter. November and December, 1980, issues are sold out. January and March, 1981, are in short supply.

Moving? Send new address and old to *Softalk* Circulation, 10761 Burbank Boulevard #8, North Hollywood, California 91801.

SOFTALK. Volume 1, Number 8. Copyright © 1981 by Softalk Publishing Inc., 10432 Burbank Boulevard, North Hollywood, CA 91801. Tel (213) 980-5074. Controlled circulation postage paid at Los Angeles, CA. Postmaster: Send address changes to Softalk, 10761 Burbank Boulevard #8, North Hollywood, CA 91801.

ADVERTISERS INDEX

CONTENTS



DEPARTMENTS

PREVIEWS

Murmurs of May . . . The Schoolhouse Apple . . . Apple's educational foundation, held over for this theme issue . . . Educational software for school and home . . . Exec Edu-Ware . . . Home Finance Wrapup . . . Bulletin Boards in Modemland . . . and more.

CONTEST: RAINY DAY CHAIN

April's contest is a game for a rainy day—and the longer it rains, the more likely you'll be to win the sunshine of $100 of goods made by our April advertisers.

The object is to make the longest possible unbroken chain of well-known names or phrases; each phrase must link by association to the ones before and after it. There are two ways to link:

1. A following phrase can begin with the word or words ending the phrase before it; for example, Space Eggs/eggs and bacon/bacon, lettuce, and tomato/tomato sauce/Saucer War/War and Peace/Peace March/March King/King Henry/Henry James/James Brothers/Brothers Four/four-in-hand/hand-me-down/down to the last drop/drop the ball/ball game/Game of the States/States Avenue/Avenue of the Americas/Americas Cup/cup and saucer—Oops, that's the end of that one, because the only linking word that can be repeated as a linking word is one that links the end to the beginning.

2. A phrase can connect to the one before or after it by association of any sort. Example: Marilyn Monroe/James Monroe/James Madison/Madison, Wisconsin/Wisconsin cheese/quiche Lorraine/Laraine Day/Leo Durocher/Brooklyn Dodgers/Los Angeles Dodgers/California Angels/Orange County/apples and oranges/Apple Computer/Hal the computer/2001: A Space Odyssey/Star Trek/Mr. Spock/Dr. Spock/raising babies/Pretty Baby/Marilyn Monroe. This one came full circle, the way a good chain should.

You can combine the two, as the second example does.

There are several ways to score points.

1. You get five points for every entry in your chain.

2. You get fifteen points extra if the end links to the beginning.

3. You get fifteen points bonus if all the links in your chain are of one kind, as in the first example.

4. Here's the kicker—you get ten extra points for every entry that refers to a *Softalk* advertiser or a product of a *Softalk* advertiser, but this relationship may *not* be used as a link.

5. You get five extra points for especially clever associations (puns allowed—if they work). Where these points are awarded is solely the prerogative of *Softalk*'s staff. If your reference is very esoteric, better explain it—briefly.

6. Total your score according to these rules and write the total on the entry form. Be sure to be accurate.

7. Deadline for entries is May 15, 1981. Enter as many chains as you like.

If it doesn't rain soon—sunny day chains are acceptable, too. Have fun.

Send your entry and form to Softalk Chain, 10761 Burbank Boulevard, #6, North Hollywood CA 91601, By May 15, 1981.

My chain is attached. If I win, the prize I'd like is:

Name:

Address:

City/State/Zip:

Your Dealer:

Win 2 weeks at Computer Camp!

Computer Camp this year meets at Zaca Lake, thirty miles north of Santa Barbara, California. Four two-week sessions are scheduled, starting on July 5, July 19, August 2, and August 16.

Campers spend approximately five hours a day learning to program. Optional outside activities include lake swimming and canoeing, square dancing, guest speakers, and campfires. Most lunches will be picnic style, combined with hikes and nature walks.

Director Garry White expects thirty to forty campers per session—an average of six or seven per counselor and two per computer. The cost is $795. A refundable $100 application fee must accompany your reservation, with $300 due upon acceptance and the balance payable by June 1. There is a discount for more than one child per family.

For further information and a brochure write Computer Camp, 1235 Coast Village Road, Suite G, Santa Barbara, CA 93108, or call 805-965-7777.

How Would You Like Computer Camp?

Softalk and California Pacific Computer Company are each offering a scholarship for one to Computer Camp this summer, in the session you choose.

Who's Eligible?

1. Entrants must be at least ten years old but no older than fifteen by June 1, 1981.

2. Entrants must be Apple owners, or children of Apple owners.

Here's How You Apply

Write an essay telling why you'd like to go to Computer Camp and what you'd like to do with a computer.

Don't Get Disqualified!

The rules:

1. Essays must be at least 200 words long but no longer than 500 words. That's one or two pages double-spaced on a typewriter or printer.

2. Essays must be computer printouts or typed or very neatly handprinted.

3. Put your name in the top right-hand corner of every page.

4. Fill out the entry form at the bottom of the page.

5. Make sure that your parents, parent, or guardian fills out their part of the form and signs it.

6. Mail your entry form and essay in time to arrive at *Softalk* by May 15, 1981.

Remember, essays that we can't read will be disqualified, so be sure yours is neat. Entries that have not had the second part filled in by a parent or guardian cannot be accepted. Be sure to follow all the rules.

Winning

Each of this year's two winners will be entitled to one two-week session at Computer Camp with tuition of $795 paid by Softalk or by California Pacific. The two winning essays will be published in *Softalk*.

The scholarships do not provide for transportation to and from the camp nor for any expenses incurred in preparing for camp. If the campers are to arrive via public transportation, Computer Camp personnel will meet them.

The Judging

Essays will be read and evaluated by *Softalk*'s editors and California Pacific's staff. Winners will be entrants whose essays excel in thought, sincerity, originality, and validity of reasons. The judgment of *Softalk*'s editors and California Pacific's staff will be final.

Attach this form to the front page of your essay and mail to *Softalk* Campership, 10761 Burbank Boulevard, #6, North Hollywood, CA 91601.

For the entrant to fill in:

Name:

Address:

City/State/Zip:

Age and birthdate:

Have you used an Apple? For how long?

Your autograph:

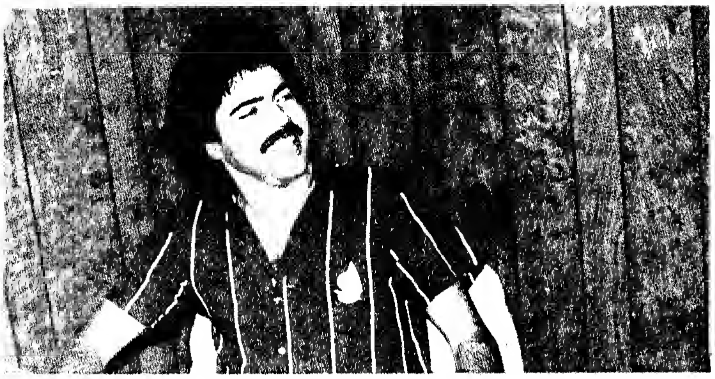
For the parent/guardian to fill in:

Name:

Relationship to entrant:

Your signature:

NASIR



BY ROBERT KOEHLER

Nasir has a considerable number of people to answer to.

Those who stay up far past their bedtimes, bound and determined to dock their ships after successful elimination of the last strike from the drone raiders.

Those who are much chagrined at bombing the hospital, thus losing all their points, when they intended to wipe out the enemy's military headquarters and get the chance to dogfight with one of their five phantom jets.

Those who grumble back at—and sometimes scream at—their Apples when the last fuzz ball, after hatching from its space egg, refuses to bounce back out into space and instead slides inexorably across the bottom of the screen, toward your ship and its complete annihilation.

For all the fury, fusillades, and violence, all this frustrated activity is the result of an innocent-enough addiction. These harried bombardiers and space pilots are actually at play with their Apples and the games Nasir Gebelli has written for them.

Still, as much joy as anger can be had from these consuming diversions, exemplified by the presence of three of Nasir's games—*Phantoms Five, Cyber Strike,* and *Star Cruiser*—in *Softalk*'s March Top Thirty poll.

Nasir can't answer for any of the emotion or attention his work has caused. In fact, he is a bit bemused by the entire phenomenon, the growth of which he is doing nothing to dissipate: his *Space Eggs* just recently appeared, and, very soon, his *Pulsar 2* should be the talk of Nasirenes everywhere.

The British novelist Graham Greene once counseled aspiring writers and all artists not to "simply steal, but steal well—steal with flair and style." Nasir, like many game programmers, instinctively takes this advice to heart. And how he borrows partially explains his popularity as a gamemaker.

Those who frequent the computer arcades in the Sacramento area—where Nasir's publisher, Sirius Software, is located—know his face by now, for this is the territory he roams, looking for the games that pique his graphical and programming interest as well as the involvement of the hordes of kids who teem the arcade halls every night of the week. Arcade fans can spot the schemes and touches Nasir has adapted from the games that met his high standards—and caused long lines of youthful players waiting to plug in their quarters and test their skill.

You only have to shell out once for a single Nasir game, making it even more attractive than its inspiration. But the star programmer wants to give the user even more: instead of a single game, he's in the process of combining two separate games on one disk that, unlike his earlier *Both Barrels*, carry the player's score forward from one game to the next, allowing complete freedom of choice in skill level.

Even better, one can say good-bye to those arcade crowds, cashier stands, and noise machines.

Like all crafted entertainments, though, Nasir's concoctions are the products of thoughtful planning and consideration. He does not write copious notes while programming. Rather, he claims, "Ninety percent of the work involved in realizing the game on screen is in my head. Virtually all my ideas are worked out before I commit the work to disk."

Which isn't to say that Nasir constructs an unmalleable master plan that programming at the Apple doesn't change. There's an evidently playful spirit behind worlds inhabited by such things as bouncing fuzz balls, and it's there when those worlds are being created. He likes to fiddle with ideas, try variations on them, see if—the ghost of impossible perfection rears its head—they can be improved.

"I'm never satisfied with what I produce. Never completely satisfied, that is. Of course, the pleasure and accompanying satisfaction that comes from seeing how my game excites and involves others is undeniable. But I know that I can always do better." He says this with the insistence of the creative artist unwilling to rest on his or her laurels.

When a project is in full throttle, Nasir spends nearly as much time drawing as writing code. Drawing, he feels, keeps his mind open to new possibilities that may enhance the game beyond its planned framework. Thus, his work habits prevent his settling into a predetermined manner of executing a game, he believes. They also help in his developing current interests, such as animation.

"Only when I see the images on screen," he explains, "can I be sure that my ideas are workable. I might have been sure that this creature or that ship was exactly as I desired—but they were on paper, not on screen. That's the real testing ground. And as I fiddle with them, they might change into something that I wouldn't have thought of in the rough draft stages."

This element of surprise has shadowed Nasir all along, from the first day he sat down with an Apple. He came to it out of near desperation, caused by an uninterrupted series of disappointing encounters with other micros. They simply wouldn't let him do what he wanted to do: input machine language code and see immediate results.

The surprise he felt when the Apple did what, for it, is a very simple task paralleled the surprise of seeing his very first program run without a hitch. The Apple's instant results spurred on his plans to devote time to assembly language programming; the results since have done nothing to alter his original opinion of the machine. He bought one only a year ago—a testimony to his prolific nature.

His facilities and skill have transferred the sensation of surprise from the programmer to the user. *Space Eggs*, especially in advanced stages, echoes the shell game, the classic game of hazard and chance. No accident this, since Nasir's original conception was of a cosmic shell match. Those floating globules were still shells until Jerry Jewell, president of Sirius Software and coauthor with Nasir of the *E-Z Draw* graphics package, looked at them.

Then, yet another surprise: Jewell, indulging in nonpresidential word antics, saw eggs, then spiders, lips, and—the one that turned Nasir's head around—fuzz balls. "It had gone," he recalls, "from weird, to weirder, to weirdest. Yet it's the only game I've written that I continue to play, because it's so unpredictable. That is probably one reason why people are so compulsive with it. I was both pleased and a bit scared when I witnessed the sight of my old roommate shooting at those shells—I mean eggs—for six hours straight."

Not all Nasir's imaginative energy is devoted to games. Like many graphics devotees committed to the Apple, he would like to see the quality of on-screen resolution significantly increased. Unlike others, however, he has some specific notions in this regard. He is convinced that the key to higher quality rests with the capability of flipping the two pages of graphics at a sufficient speed to double normal resolution and eliminate the flicker and flashing one now encounters on the Apple. Some experimenting with this—you can be sure that Nasir and Sirius are keeping their eyes on its progress—is ensuing. Results aren't here yet, but Nasir and friends are confident that, soon, the resolution problems will be indeed resolved.

This same feeling carries over to Nasir's view of the future. "Virtually everyone," he speculates, "will have a computer. People generally don't trust electronics and computers; anything like the Apple that personalizes use breaks down that sense of distrust. But teaching kids with computers in school is going to really turn it around. The school of the future might be a central computer bank with students at home plugging into it via modem. I see kids now whose computer aptitude at a fairly young age is remarkable. My knowledge of the Apple has its limits, and if I had had the kind of instruction kids are and will be receiving, I'd be far ahead of where I am now."

With a new game, better than its predecessor, to conquer every couple of months, Nasirenes sifting through their growing mountains of disks and trying to decide which one to run next must be boggled by their favorite gamemaker's self-criticism. It is the stuff that makes up the mind-set of a master craftsman.

"Do you know what I'd really like to do?" Nasir asks.

"I'm amused and a little upset by computerists who say, when their project has hit a seeming dead end, that it can't be done. They aren't thinking and aren't pushing themselves enough. I'd like to be a troubleshooter on those follies—or what they think are follies. The same kind of creative intelligence that developed the idea of laser disks, making for a veritable library of graphic slides that might eventually allow you to make a small movie on your Apple, can combat problems that others have given up on."

Nasir and Phil Knapp, general manager of Sirius Software, watch as Jerry Jewell, Sirius's president, tries a module of Pulsar 2.

Even charting drone strikes and fantasy schemes in outer space would offer no higher challenge to the programmer. And even if Nasir does bid farewell to space eventually, you can be sure that the Dog Star is smiling down at him tonight.

OPEN DISCUSSION

Supercomputer
Softalk is a great idea. Keep us all together. Buying an Apple is like finding a baby on your doorstep and then two weeks later discovering that his name is Clark Kent. I'm doing things on mine that I used to do on an IBM 360.
H. Speer, Sea Cliff, NY

Distribution Does Not a Publisher Make
Your "Exec On-Line Systems" article zeroed in perfectly on Ken and Roberta Williams and many of the reasons for their success. I've been exposed to Ken's technical talents since working with him at Informatics and he unquestionably ranks with Bill Budge, Bob Bishop, Nasir, and just a very few others as one of the software greats of our industry.

In general, the article was very accurate; however, there was an oversight that needs to be clarified. The *VersaWriter* that Ken used for *Mystery House* is a product of Versa Computing Inc., not Peripherals Plus as you indicated. Our company, Softsel (formerly Robwin Computing Corp.), is a distributor of *VersaWriter* as is Peripherals Plus. In fact, Versa Computing is just now releasing a new software product called the *VersaWriter Expansion Pac-I*, which will greatly enhance the *VersaWriter*'s capabilities. Thanks for the opportunity to clarify the situation.

Softsel wishes *Softalk* continued success. You're off to a great start.
Bob Leff, President, Softsel, Marina Del Rey, CA

More on the Remarkable Williamses
I very much enjoyed the profile of Ken and Roberta Williams in the February '81 *Softalk*. I have the *Paddle Graphics* program from On-Line Systems. It took several telephone calls to Coarsegold to order the software. Either Ken Williams or I seemed to be out of our respective offices. (No reflection on Coarsegold or Santa Barbara.) The phone conversation closed with Ken Williams's advice, "If you have trouble with the program, call me. You're doing something wrong." That's confidence in your software! I've had no reason to call for help—three a.m. or otherwise.

I enjoy your publication and have ordered several products from your advertisers. Keep up the fine work.
John A. Reyburn, Jr., M.D., Goleta, CA

Lemon Joins Apples for Softalk
Best reading for the Apple II owner. As founder of Apple Corps of San Diego, I suggest every owner should read *Softalk.*
Philip A. Lemon, San Diego, CA

Poet's Great, but Softalk's Late. Ah, Such Is Fate!

There's a magazine called Softalky
Whose deliv'ry is a bit balky,
So I'm not delirious,
Just downright Sirius,
For I might have won in a walky!

Unfair to East Coast!!! There is no way we can compose a sensible entry to a contest with a February 2nd deadline when the magazine only arrives on January 31, and this is not the first time this has happened. The stores have them here a week before I do, but it does not seem fair to read theirs when I am not going to buy one. Help!
George S. Forde, Philadelphia, PA

Thanks for a delightful limerick! Indeed it was too late, and we apologize for the mail. However, a large number of the semifinalist limericks are from readers in eastern states—New Jersey, New York, Virginia, Massachusetts, for example.

We are trying very hard to straighten out the mailing problems. Many people on the East Coast get Softalk *in a timely fashion and a few people even in California still receive their copies very late.*

Ever Kept House for a Family?
I thought that R. L. Coleman's projection of the use of the microcomputer for housekeeping tasks was little short of ridiculous. I can't think of a more trivial use of a microchip than to open a garage door, when all one has to do now is press the button on a small transmitter to accomplish the same task. And computerized menus? What for? Suppose I don't want what the computer dished up for me—maybe my stomach is off, and I prefer scrambled eggs tonight. And how about my wife, who happens to like dreaming up fabulous new meals? And two sixth graders working on problems in "advanced computer programming," what problems could a sixth grader give to a computer? Finite element analysis of stress in a bridge girder? List processing applied to artificial intelligence? It's more likely a new version of NIM. Big deal!

I think the home computer can be a fine learning tool. I can see its use as a "talking book." It might considerably reduce the cost of a home encyclopedia, and permit learning tools better adapted to the particular interest of the child (provided the software is available). It could provide home drill in school subjects. I can see it as a useful device for making electronic mail a reality. And certainly

children can learn the elements of programming just as they learn the elements of arithmetic. I can also understand that, in learning to program a computer, a child could grasp more easily the requirements of logical thought. This is the direction in which home computer applications should be focussed.
John Figueras, Victor, NY

The children in Coleman's imaginary family were using the computer to tackle a problem—how to program the computer. And, if those kids were developing a new version of NIM—what an achievement that would be for many of us. An exercise, perhaps, but a perfectly legitimate learning tool.

The next letter indicates that you may be in the minority.

I'm writing to say thanks. After seeing the articles on the computer-run home and the shopping Apple and hearing about the response of people from across the country, the dean of my college issued an "order" to me to get to work on the [Louisiana Tech] microcomputer laboratory article I promised you.

The response to the article on Menu Planner is unbelievable. As of yesterday, I have received letters wanting more information about the program from twenty-six different states ranging literally from Maine to California and from Florida to Oregon. Incredible!

Thanks again for everything.
Dr. Rick Coleman, Ruston, LA

World Travel for Right Programmer

It's about time someone began trying to tie together all the loose Apple projects bubbling out all over the United States and present some type of a periodical that would be of interest to Apple owners worldwide.

I own an Apple Europlus II, two disk drives, an American (Toshiba) color TV being used as a video, and a Centronics 703 printer. I've connected my Apple to the TV with a SUP'R'MOD II interface unit manufactured by M&R Enterprises, Sunnyvale, CA, but still have no color graphics. Can someone tell me the solution to this problem?

I own an H&R Block franchise over here in Germany (actually two of them—one in Frankfurt and one in Schweinfurt) and have created a new corporation in the United States. We also sell automobile insurance to the Americans stationed in Europe with the armed forces and provide for them a variety of other services such as pet shipping and automobile shipping. I have purchased a tax package from a company in Milwaukee and will soon be doing all my tax returns over here via Apple and probably independently at that.

In this regard, I employed a programmer in the USA to assist me with a project, but getting any response from him is very difficult. What it is about motivation and Americans I'll never figure out, but I'd like to ask your help and assistance with the project. My programmer has begun adapting my insurance program and methods of transacting business to diskette but he has still been unable to finish the project and furnish me with something that will function properly.

So, here's my offer. I'll buy the ride (round-trip air fare for one Delta Air Lines from the United States to Frankfurt, Germany) this summer (late August, early September) or later if desired to the person who can automate the bookkeeping process I have to contend with through my employer and who can finalize and complete the insurance sales package I'm trying to market. You figure out how to promote it and it's yours. I'll supply written copies of what I have now and what I need in generalized form and you tell me what else I need to do.
Michael Screeton, Frankfurt, West Germany.

Send your qualifications, tips, or requests for more information to Michael Screeton, Am Dornbusch 33, D6000 Frankfurt, West Germany.

Toward More Challenging Games

We would like to make one suggestion to the producers of games. That is, they supply the purchasers with written instructions on how to make variances to the game after it becomes repetitious. I refer primarily to *Apple Invaders*. We have no difficulty in obtaining the maximum score of ten thousand per game. Therefore, we would like to increase the maximum to make the game more challenging. We have been playing a game called *Sink the Sub*. This is a time-elapsed game but now we feel it runs too slowly to be interesting and, therefore, would like to increase the speed of the game.

Also, is there a source that one may contact for assistance in altering existing business programs?

We purchased the Apple II primarily to handle our business accounts. We have examined several business programs and find the one most suitable to our needs is the *File Cabinet* but with a few necessary alterations. For example, we wish to utilize the *File Cabinet* program for inventory purposes but to select the inventory from any combination of from one to ten columns, and, instead of horizontal totals, we require products [multiplication].

If such a programming aid source is available, please advise. We would certainly be willing to pay for such a service.

Of course, we have been continually studying the reference manuals that we received with the Apple II so that at a later date we may be able to make our own program adjustments but, until such

GOTO 70

Super Invader Is Readers' Choice

The biggest surprise about the results of *Softalk*'s Most Popular Program Ever poll was that there were no surprises.

Of the top ten programs, as chosen by ballot by the readers of *Softalk*, only one, *Apple Writer*, has not made the top ten in *Softalk*'s bestseller list at least once.

Super Invader, the Japanese import written by M. Hata and sold here by Creative Computing and California Pacific, was the choice—by a narrow margin over Microsoft's *Adventure* and Personal's *VisiCalc*.

Others in the top ten, in order, were *Sargon*, by Dan and Kathe Spracklen, Hayden; *Asteroids in Space*, by Bruce Wallace, Quality Software; *Flight Simulator*, by Bruce Artwick, SubLogic; *Hi-Res Adventure #2: The Wizard and the Princess*, by Ken and Roberta Williams, On-Line Systems; *Odyssey*, by Robert Clardy, Synergistic Software; *DOS 3.3*, Apple Computer Inc., and *Apple Writer*.

Tabulations were made on the basis of ten points for a first-place vote, nine points for second, and so forth down to one point for a tenth-place vote.

Tightness of the balloting in the first three positions is indicated by a margin of only one point for *Super Invader* over *Adventure* and only six points for *Adventure* over *VisiCalc*.

Invader was listed on many more ballots than any other program, but it actually received fewer first-place ballots than many of the programs in the top ten.

Adventure, by Software Associates/Gordon Letwin, missed the top spot by the narrowest of margins, gathering more first-place ballots but slightly weaker overall support.

In contrast, *VisiCalc*, Personal Software's product from Software Arts/Dan Bricklin and Robert Frankston, received the most first-place ballots of any program, but was listed on far fewer ballots than any other program among the top ten. It's a real tribute to the revolutionary nature of *VisiCalc* that it could compete on equal terms with game programs in a popularity poll when its essence is serious business applications.

No individual author placed more than one program in the top ten and only Apple Computer Inc. published more than one of the top ten programs.

Second to *VisiCalc* in first-place votes were *Hi-Res Adventure #2: Wizard and Princess* and *Sargon II*. Fourth in first-place votes was *Dogfight*, but it did not get the breadth of support necessary to make the top ten.

Diversity of interests of Apple owners is indicated by the fact that one hundred seventy-eight different pieces of software were cited on the ballots.

Narrowly missing the top ten—in fact tied for eleventh only one point behind *Apple Writer*—were *Bill Budge's Space Album*, by Bill Budge, California Pacific, and *Temple of Apshai* from Automated Simulations.

In thirteenth was *Hi-Res Adventure #1: Mystery House*, by Roberta and Ken Williams, On-Line Systems, followed by *Cyber Strike*, Nasir, Sirius Software, and *Easy Writer*, by John Draper, Information Unlimited. *Dogfight*, by Bill Basham, Micro Lab, tied with *Easy Writer* for fifteenth.

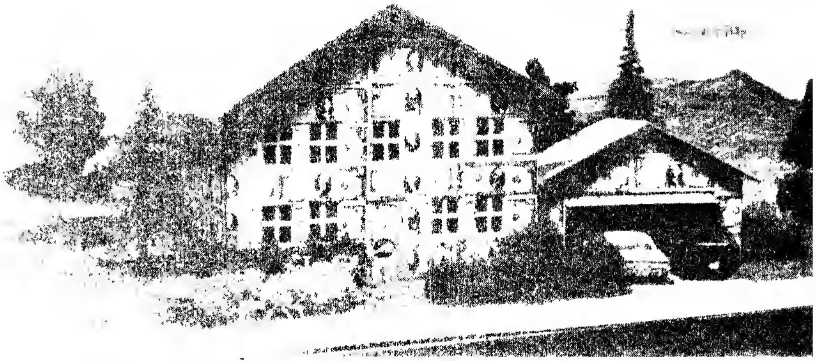
Readers with a programming bent selected *DOS Tool Kit* from Apple Computer Inc. as their favorite programming tool with *Apple-Doc*, by Roger Wagner, Southwestern Data Systems, a clear second.

In the category of business software, *Apple Plot* from Apple Computer ran next behind *Easy Writer* with *CCA Data Management System*, Creative Computer Applications/Ben Herman, Personal Software, as the leading data base.

Among graphic utilities, *Bill Budge's 3-D Graphics System*, by Bill Budge, California Pacific, led by a single point over *Apple World*, by Paul Lutus, United Software of America. Lutus, incidentally, authored the original version of *Apple Writer*.

M. Hata, as author of the most popular program, will receive a trophy symbolic of the approval of Apple owners. The award is being presented at the West Coast Computer Faire early this month.

Next year's contest will be restricted to programs that were released between November 1980 and November 1981, resulting not in a most popular program ever decision, but in a most popular program of 1981, as chosen by *Softalk*'s readers.



Where Does the Money Go?

BY CRAIG STINSON

If you've ever accidentally buried your checkbook under a pile of computer magazines and then agonized for days about how much money you didn't have and which checks should be reported lost, you've probably realized that one of the most practical things you can get for your Apple is a home finance package.

You may have discovered also that choosing the right package is not a simple matter, because there is a great variety of such programs available, at a great range in cost. Which program is best for you will depend not only on the complexity of your needs and the amount you're willing to spend, but also on what particular things you want it to do for you. Almost all these programs have some unique feature or features of special importance to someone.

Other factors that may affect your decision are the extent of your equipment—whether you have one or two disk drives, for example—and the degree of your computer sophistication. Certain programs will prompt you every step of the way; others require at least enough understanding of Basic to customize certain lines in the program.

Double Article on Single-Entry Systems. This is the first of two articles surveying the plethora of home finance software. The packages discussed here should not be construed as superior or inferior to the ones to be covered next issue, nor are they in any other way logically antecedent to them. They simply were available sooner for review.

The spread in terms of capability between the least and most sophisticated of these finance programs makes categorization difficult. At the upper end of the range, it's a bit hard to draw the line between a home finance program and a small business controller. Our common denominator, however, is that all programs considered here will be single-entry accounting systems. Some of the more fancy of them could indeed be applied to small business requirements as well as to managing the family exchequer.

Checkbook from Programma. Steven Welch's *Checkbook*, published by Programma International (Burbank, CA), requires 12K RAM and a disk drive and sells for $34.95. Its documentation boasts a program so compactly written that the user can keep large amounts of data—probably a year's worth—on the same disk as the software. That's a nice feature if your needs are modest, because it saves you the trouble of switching data and program disks while you work. Presumably you will want to copy your data onto separate disks for backup and archive, but that doesn't need to slow you down during the data-entry process.

Programma's main menu displays the register number—you can run several accounts on the same disk, if you like—the number of transactions recorded to date on that register, and the current balance. It offers options of posting checks and deposits; listing the account to the tube, printer, or both; searching for particular items; quitting; and reconciling your accounts against a statement.

The data base offers five fields: transaction number, date, payee, memorandum, and amount. The payee and memo fields hold up to twenty characters apiece, and the memos can be used later for sorting checks into budget categories, although there is no specific module in the package for establishing budget levels and comparing actual to projected amounts.

The system distinguishes checks from deposits by looking at the payee line. If the first character there is a number sign, the entry will be construed as a deposit and the amount will be credited to your account. All other entries are debits.

A useful feature is that the transaction number need not be strictly numeric. So, for example, if you get money out of your account through an automatic teller window, without writing a check, that withdrawal can be coded with a W or some other character. Likewise, a telephone transfer or automatic monthly payment can be recorded with some specific code—as well as a number.

When you finish inputting the data for a particular item, the program writes it off to the disk; you do not have the opportunity to correct it at that point. Later, you can edit entries by way of the listing option on the main menu.

Don't Bury the Disk, Too! Three modes of listing are available: all entries, date to date, and transaction number to transaction number. Unlike most of the other systems, the Programma *Checkbook* does not organize data into monthly files or even yearly ones. It simply keeps a check register, much like the one you buried under those computer mags. So the all-entry listing does indeed give you the whole thing, a full screen at a time. You can stop the output at the end of any screen, however, and if you spot an item that needs fixing, it can be called up via a system-generated index number and amended. The list can be sent to the printer a page at a time, as well.

When you ask for a listing, by whichever mode, the entries appear in the order of ascending index number; in other words, in the order you entered them. If you post your checks out of order, they will remain that way.

The search module offers seven options: outstanding checks and deposits, specific date, payee, memorandum, item number, amount, and index number. If you use the memorandum line for budget categories, the search feature will display and total all entries for a given budget category for a specified time period. This is at least half of what most people want out of a budget program. Of course, then you can't use the memo as a true check memo; you probably wouldn't want to record *necktie* when the check belongs to the category *clothing*.

HLP AT TX TM. A nice feature of the search routine is that you can specify the beginning characters of a payee or memorandum line alone and use them as a template. So you could prefix any tax-deductible budget category with a TX and accomplish a search and summation of all deductible transactions as well as a separation into individual budget categories.

The reconcile program displays each outstanding item individually and asks you if it appears on your bank statement. Once you answer affirmatively, that item appears in subsequent listings with an asterisk and does not appear in subsequent reconcile queries. After you finish this procedure for all outstanding items, a separate spot on the main menu asks you for the balance on your statement, checks to make sure you've entered all service charges, scans your file for outstanding credits or debits, and tells you how far off you are from the bank.

Documentation for the Programma *Checkbook* is lean but adequate. What isn't obvious from the four pages of printed description is usually made clear through prompts in the program itself.

Spectrum's Budgeting Checkbook. *Check Register and Budget* from Spectrum Software, programmed by Andrew Thompson, offers roughly the same degree of data capture as the Programma *Checkbook*, with the addition of a specific cost/budget analysis routine, some enhanced search and sort capabilities, and a simplified data entry method using predefined purpose and recipient categories. The program requires Applesoft, 32K RAM, and a disk drive. By itself it sells for $39.95; a complete *Home Finance Pak I*, comprised of this and noninterfacing companion programs for tracking savings accounts and credit cards, goes for an additional ten dollars.

When you initialize your Spectrum checkbook, you first specify the current calendar year, then you're asked to list up to thirty-two twenty-character standard check purposes and check recipients. Actually, the first two of these in each column are reserved for deposits and miscellaneous checks respectively, so you really have thirty standard categories at your disposal. That's enough to cover the vast majority of most people's transactions; and, when you do need to post an item into the miscellaneous category, you get to spell out the particular purpose and/or recipient for that item in twenty characters or less.

So there is really no constraint on the number of categories or recipients available; the system simply allows you to post recurrent transactions with single numbers instead of having you type out the whole data field each time. There are advantages to this method in terms of disk storage space and ease of data entry. Since purpose and recipient fields will later be used as search criteria, you don't need to worry about keeping your spelling consistent or about whether you've been recording supermarket purchases as food or as groceries.

As you establish each item on your list of standard purposes, you are asked whether it is to receive a tax flag or not. This approach produces roughly the same level of eventual search and sort capability as Programma users can get by prefixing their memoranda with TX or some such code. It doesn't really amount to an independent data field, but you can produce lists of all tax-coded items as well as of individual budget categories.

If at any point during your use of the program you wish to add or alter standard purpose or recipient categories, a special item on the main menu allows you to do that.

The Bad News Monthly. Like Programma's *Checkbook*, the Spectrum software is compact enough to allow data storage on the same disk with the program. Unlike Programma, however, Spectrum keeps all files in monthly units, which offers distinct advantages for searching, listing, and establishing budget amounts. In the budget module, for example, you can enter occasional expenses like insurance premiums or vacations into the appropriate monthly files, or you can specify a more constant kind of expense for any given month and ask the system to write it into all twelve files.

When you dial the cost/budget report module from the main menu, the system first asks you to specify what month or range of months you wish to examine, then it brings up that predefined list of standard check purposes (including number one for deposits and number two for miscellany) in a tabular format that displays total expense (or income) per category, budgeted amount, and positive or negative balance. Should you wish at this time to bring your budget into better alignment with the reality of your check register, simply strike A and the purpose category and key in the new amount. Or you can hit a P and send the whole thing off to your printer.

So much for budgeting. Check and deposit posting is made neat and simple by the predefined category approach. There are a couple of disappointments here, however. First, the data entry module does not display your current balance, so you've got to go to a separate "Review the Check Register" module to see whether you dare pay that last bill or not. Second, there is no provision in the system for transactions with your account that are neither checks nor deposits. Since automatic window withdrawals, telephone transfers, and interest-bearing checking accounts are becoming fairly commonplace, this seems a disconcerting limitation. You can enter such things, of course; you just have to mickeymouse the number—use a negative number or something clearly out of the range of your current register.

Easy Search Is Hard To Reconcile. On the other hand, there are some nice touches in terms of search capability. Besides being able to search and list checks by check number, purpose, recipient, and the presence of a tax flag, you can also seek out all transactions occurring on a particular day of the month during any range of months you desire; and you can search for transactions that fall within a given amount range. Any check or deposit of more than five hundred dollars, for example, could be called up with a single search request.

The reconcile routine is a trifle less automatic than that of the Programma system. Rather than being led through all your outstanding entries one by one, you have to scroll through the check register yourself and note anything not yet cleared by the bank, then enter those items into the separate reconcile module.

The Check Register and Budget comes attractively packaged. Documentation is written in English—a fact not to be taken lightly—and the presentation of material through the program itself is such that one could comfortably recommend it to a novice computer user. The checkbook review option, for example, presents a display of one's check register that probably comes as close as possible to looking like that old-fashioned thing you stick in your purse or back pocket.

The optional companion programs in the *Home Finance Pak I, Savings* and *Credit Card*, are relatively simple items that post transactions and record balances for as many accounts of each kind as you desire. They will generate printed reports, as will any of the options in the checkbook program, but they do not provide search and sort capabilities and they do not interact with the budget list generated in the checkbook program.

Continental's Financial Sentry. The major step up in accounting capability offered by Continental software's *Home Money Minder* is that it does interface cash and credit card transactions with the overall budget plan. It also allows you to write checks or make cash payments to a credit card account and have the amounts posted automatically on the separate credit card file.

The system, programmed by Bob Schoenburg, requires Applesoft, 48K RAM, and at least one disk drive. Two drives will make your life a good deal simpler, but the system will configure for one and will prompt you for all necessary disk changes. The price is $34.95.

Besides being more powerful, the *Home Money Minder* is a bit more cumbersome to use, especially if you don't have that second disk drive or a 132-column printer. Again, as with the two systems described so far, you can certainly keep your records without the printer, but when Continental sends a list to the tube it sends out that same 132-column data in such a way that lines wrap around on your screen and get clumsy to read.

So your first consideration, perhaps, should be how much accounting power you really want. If you want to track everything you spend and take in, including all your cash transactions, and you have the requisite hardware and patience to do it, then this is definitely an economical way to go.

Here's how it works. First stop after initialization is the budget menu, where you can establish as many as fifty categories, at a maximum of fifteen characters each, and specify either a common amount per month or separate amounts for each month of the year. Income categories are flagged to the system using an asterisk as a precedence code. Having created the budget, you can then print it, review it, correct it, add to it, and print it again.

You will want to return to this budget menu occasionally for additions or revisions or to get actual versus projected expense and income reports. The reports are available through the printer (or screen if you don't have a printer) or through a CRT graphing module that gives you a month-by-month bar graph display, showing actual amount in one color and projected amount in another. The system will not generate a report or a graph for any given month, however, until that month is actually closed out.

Fodo Will Feed You When Groceries Won't. This brings up another important general point about the *Home Money Minder*. A number of features are built in to reduce the likelihood of erroneous data entry. For example, if you should try to post a check out of sequence, you'll get a beep and a polite inquiry about whether you really mean to do what you're about to do. The system also discourages you from entering checks or deposits into the wrong month.

Unlike the Spectrum register, *HMM* does require you to spell out your budget category with each entry. If you misspell a category or present it in a slightly variant form, it will honk and suggest the closest match it can find to the categories as you entered them in the budget routine. Type "fodo," for example, and it will suggest "food." But, if you hit "groceries" when your budget is for food, you will most likely get a message saying the system can't match that one, and do you want to retype.

Back to the main menu. Having made your heading and filed your budget, your last option (besides exit) is income and expenses, and from here you branch into checking account, credit cards, or cash, depending on what sort of data you wish to enter.

The data bases for all three kinds of transactions are in six fields. For checks you specify check number, date, payee, amount, budget category, and memo. If you specify credit card for the category and name the card in the memo, the system will update the appropriate card account. If you write a check for cash and enter cash on the category line, the amount will be transferred to the corresponding monthly cash account file.

If you don't use the memo line in this fashion to effect an interface with another account, you can use it as a genuine check memorandum. Up to ten characters are available to remind you where your money went.

For deposits, the fields are deposit code, date, source, amount, category, and memo. Transfers to your checking account from a credit card account can be handled the same way as payments. The deposit code can be used either as a transaction number or as a flag for tax reporting.

Similarly, the last five fields for credit card and cash data are for date, recipient, amount, category, and memo. The first field in the credit card allows you to name the card; for cash the first field allows you to specify a code for whatever purpose you want.

Stacked for Concentration. Whenever you post any data, the system displays the six fields stacked vertically, as opposed to the columnar, check-register-like format employed by Programma and Spectrum. This has the disadvantage of allowing you to see only one entry at a time while posting data, but it does permit you to confirm the data before sending it off to disk. Type "0" if all is well or a number from 1 to 6 to change a particular field. Your current balance is also displayed at all times, in flash mode if it's negative.

The system will generate printouts of all activity in any account for any given month. It will also sort and print all activity for any particular data field item, and because of the interface between cash, credit cards, and checking, you can display or print all expenditures of any kind for a given budget category for a given month. This is really handy if you're trying to keep track of how and whither all that small coinage in your pocket trickles away.

A couple of other pluses and minuses: when searching for payee, deposit source, or memorandum, you can specify the first few characters only and the system will bring you everything beginning with that combination. Under credit cards you have the option of entering returns as well as purchases; it would be nice if you could enter voided checks in similar fashion. Unfortunately, the system still has no distinct provision for entering transactions such as automatic teller withdrawals or automatic payments.

The Bookkeeper: Competent and Personable. What's immediately striking about John Owens's *The Bookkeeper* from Delta Software (Applesoft, 32K, disk drive, $89.95) is the elegance of its design. It has a few features that the programs considered thus far haven't, but it lacks some of the power. Yet what it does, it does with a finesse that will justify the extra expense, at least for some.

First, it is extraordinarily well documented, even outlining step-by-step how to *Muffin* the 3.2 master for your 3.3 system, if that happens to be what you have. And, while the system is running, it keeps you prompted with friendly first-person messages like, "You have finished making entries; wait while I file them and we will continue. Please do not interrupt."

The process of data entry, moreover, is made about as effortless as possible by the use of standard, predefined accounts, much like the standard purpose approach of the Spectrum *Check Register*, and by a system of default options.

When you go into the data posting mode you are offered four choices: check, deposit, balance, or finished. Next to the query, "Which one?" is the letter *C* with the cursor flashing over it. If you want to enter a check—the most commonly selected option—all you do is hit return. Having done that, you see next the current transaction number (a system-generated index that identifies every entry for an entire fiscal year), the current month, your present balance, and the check number that the computer expects to see next. If that's the check number you're about to enter, again you merely hit return. Next you need to put in the day of the month and, unless this check is not the first of a given posting session, the computer will display the date of your last entry, on the assumption that all checks posted at a single session are probably going to have the same date. If this is not the case, you merely type in the correction and the computer updates its default value.

Default to Negative Keeps You Out of Trouble. The "Pay to" line displays twenty-three dashes next to the question mark, so you know exactly how many characters you may use. After you type in the payee and the amount, you are asked for an account number. Here the cursor flashes over a letter *A*, and if you hit return the transaction display disappears for the moment to be replaced by a list of standard accounts, or budget categories, that you've established during the initialization of your check register. When you find the appropriate account number, another return takes you back to the transaction display; type in the number, and the name of the account appears.

At that point, you're asked to confirm or reject what you have done so far with a Y or an N. The default reply here is N, making it less probable that you will send off an erroneous entry. If you do need to correct anything, hitting returns will take you step-by-step through the data you've entered, allowing you to fix whatever may need fixing.

Finally, you're given the opportunity to add a note to the entry. This amounts to a greatly expanded memorandum line—up to 159 characters—except that the note is written into a separate text file on the disk, under the assumption that for most entries you won't need one. Those entries that have notes will appear in all subsequent printouts with an asterisk, and the note will show up underneath as a footnote, keyed to the index number of the transaction.

So the system of default values, with the computer anticipating your most likely moves, has the dual benefit of reducing keystrokes and protecting you somewhat against mistakes; the account number system reduces disk storage and ensures uniform spelling of budget categories; and the note method of appending memoranda allows you lots of space where you really need it while wasting none where you don't.

Just in the N/CK of Time. There are a couple of other useful features to be found in the check/deposit entry module. If you strike N for check number, the system replies with "N/CK" and allows you to enter any sort of miscellaneous withdrawal-type transaction you wish, as always with the option of adding an explanatory note at the bottom. Enter VOID on the payee line, and the system will void the check and allow you to note the circumstances.

The other systems discussed so far will handle these kinds of transactions also, but not as directly. There is definitely something to be said for seeing "N/CK" on a summary printout, rather than, say, "—1."

One thing that *The Bookkeeper* does that the others do not do at all is allow for a split-account check. If, for example, you write a check for $40 worth of groceries and $25 in cash, you can enter S for account number, and the system will prompt you through a maximum three-way split, updating each account appropriately. Unfortunately, there is no similar provision for multisource deposits.

So much for easy input. There are a few disadvantages in terms of overall accounting capability relative to the other three programs considered here. You can search and display transactions by account number, but not by payee, memorandum, or transaction number. And there is neither a provision for tax coding nor a specific program module for bank reconciliation.

In compensation for the lack of payee search capability, you get to define up to ninety check accounts and nine for deposit sources. Hence, you could set the system up to cover practically everyone you write checks to in your normal course of affairs. The trouble is that then you would be trading budget convenience for search capability, because you would wind up having separate budget entries for every store you visit instead of for generic categories.

Budget projections are created only on a month-by-month basis, instead of in the larger time chunks permitted by other systems. Yet you can display cumulative actual versus budget totals in any category for any period of time you like and—unlike the *Home Money Minder—The Bookkeeper* allows you to get a budget report for the current month before the monthly file is closed.

Wired for Emergency. Furthermore, the monthly summary display not only reveals how much you are over or under your allowance, but also shows you what percentage of your actual total expenditures has been accounted to each individual category. It also displays an overall cash-flow summary at the bottom of the list to show you the direction of your fortunes for any month you choose.

For those who require privacy, the program recognizes a password you designate for entrance to the data files. If you fail the test in three tries, the Apple makes like an anti-car-theft alarm and beeps continuously until you hit the power switch. And, just in case you forget and find yourself locked out of your own system, there is a procedure to get past the sentry.

An optional *Check Writer* program, for $39.95, will interface with *The Bookkeeper*. It will print your checks for you, on forms that you can order encoded with your bank account number, and will automatically enter them into the check register portion of *The Bookkeeper*.

Stay tuned. There are several more home finance programs to be discussed in the next issue.

VENTURES WITH VISICALC

BY FRANK MALONE

Family Finance with VisiCalc

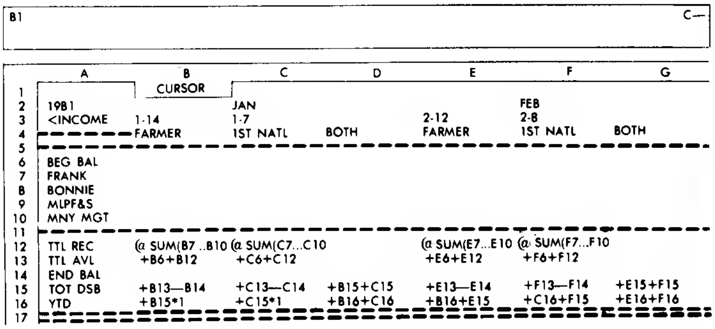
For the family of means by no means who, nonetheless, has a couple of bank accounts—one for each provider—making checks balance can be a mind boggler. Keeping track of one checking account a month is bad enough, but two. . . .

Unless you use *VisiCalc*. Right into your living room or study or office, *VisiCalc* offers the basis for a checkbook program that requires little more than turning on your favorite computer and setting up a common-sense format for yourself.

For example, if your accountant advises you to maintain records of all your income and expenses, *VisiCalc* does the hard part—the summing of items down columns and across rows—so that, at year's end, you can print a neat package of information that not only has saved you much time (and ultimately money) but also provides accurate data for tax purposes.

One Road to Keeping Records. There must be many ways to do it with *VisiCalc*. Here's one of them. And the beauty of using *VisiCalc* is that, with minimum effort on your part, you can set up and maintain your records as you see fit. Then, if you come up with better ideas as you progress, well, use them later, maybe next year.

Begin by booting up your system with *VisiCalc*, of course. Then start your entries with your bank statements. It's best to use checks for expenditures to work with as few sets of records as possible. But first, start setting up your format by typing labels on your *VisiCalc* worksheet for income for a six-month period. Figure 1 shows this format.



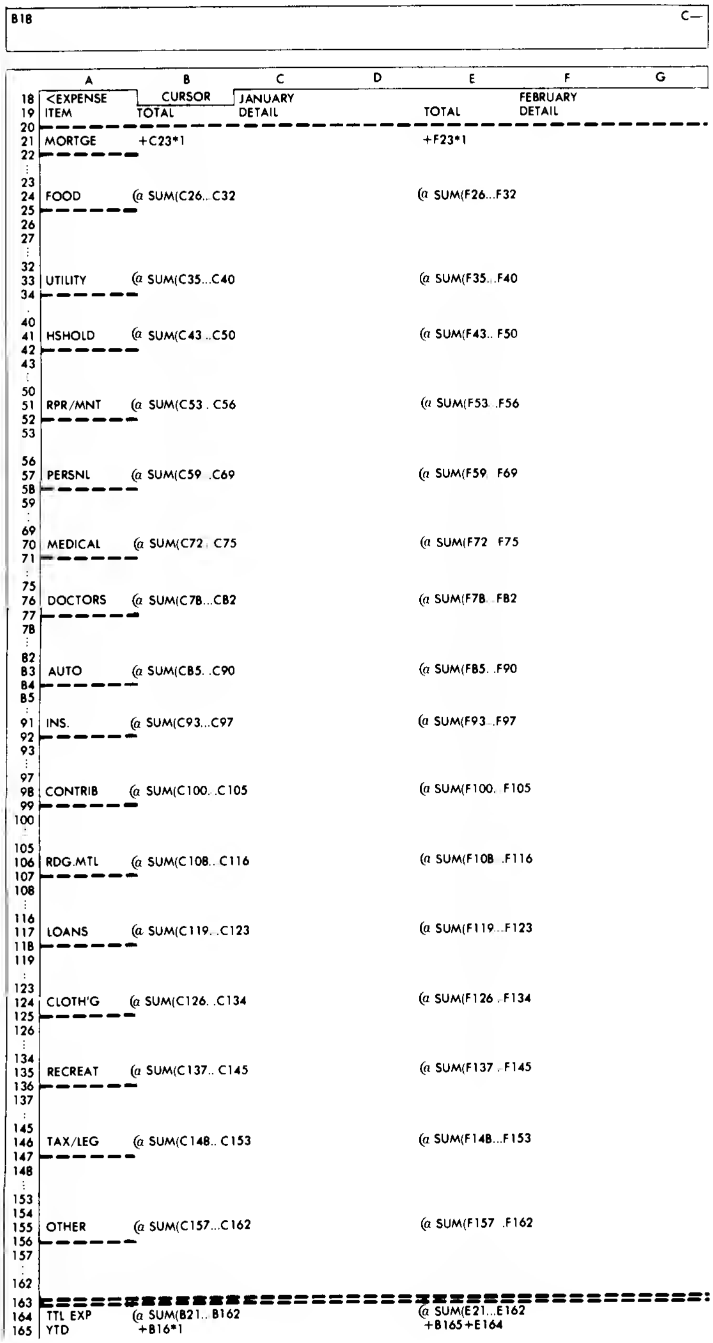
```
B1                                                                        C—

          A           B           C           D           E           F           G
 1                 CURSOR
 2   19B1                      JAN                                  FEB
 3   <INCOME      1-14         1-7                      2-12        2-B
 4   ------      FARMER       1ST NATL     BOTH        FARMER      1ST NATL     BOTH
 5   ----------------------------------------------------------------------------
 6   BEG BAL
 7   FRANK
 B   BONNIE
 9   MLPF&S
10   MNY MGT
11   ----------------------------------------------------------------------------
12   TTL REC     @ SUM(B7..B10 @ SUM(C7...C10           @ SUM(E7...E10 @ SUM(F7...F10
13   TTL AVL      +B6+B12      +C6+C12                  +E6+E12      +F6+F12
14   END BAL
15   TOT DSB      +B13—B14     +C13—C14    +B15+C15     +E13—E14     +F13—F14     +E15+F15
16   YTD          +B15*1       +C15*1      +B16+C16     +B16+E15     +C16+F15     +E16+F16
17   ============================================================================
```

Figure 1. Income formot. The farmats far incame ond expenses simulate o *VisiCalc* worksheet, but, instead of dollor omaunts, the formulas far monipulating the dallor omaunts ore entered, so that yau can see haw they are derived in their appropriate rows ond columns.

To ensure to-the-penny accuracy, enter exact amounts when you start putting in your checking account numbers. Then, if you desire easier viewing, you can use a global command to change to integers. Your original numbers will be preserved, and you can call them back at will. For example: the command /GFI changes 48.98 to 49 or 1962.24 to 1962, and the command /GF$ changes them back again.

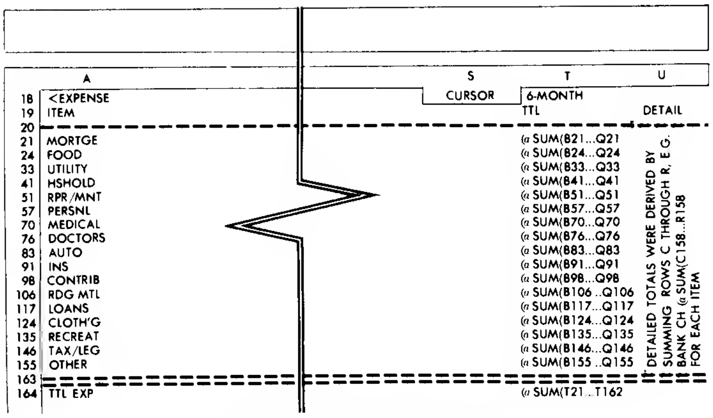
But perhaps this is jumping the gun. There are many commands that will facilitate your using, viewing, and printing your checkbook statement. Some of these will be listed here, but for an easy trip through command-land and for other valuable tips (after you've read your *VisiCalc* manual carefully) consult the reference card that comes with your manual.

Next, on your worksheet, type in expense labels down column A, underneath your income format. Then type labels for the first six months' detail. Figure 2 shows this format.



```
B1B                                                                       C—

          A           B           C           D           E           F           G
 18  <EXPENSE       CURSOR     JANUARY                              FEBRUARY
 19  ITEM         TOTAL        DETAIL                   TOTAL        DETAIL
 20  ----------------------------------------------------------------------------
 21  MORTGE       +C23*1                                +F23*1
 22  ------
  :
 23
 24  FOOD        @ SUM(C26...C32                        @ SUM(F26...F32
 25  ------
 26
 27
  :
 32
 33  UTILITY     @ SUM(C35...C40                        @ SUM(F35...F40
 34  ------
  .
 40
 41  HSHOLD      @ SUM(C43..C50                         @ SUM(F43.. F50
 42  ------
 43
  :
 50
 51  RPR/MNT     @ SUM(C53 . C56                        @ SUM(F53 .F56
 52  ------
 53

 56
 57  PERSNL      @ SUM(C59 .C69                         @ SUM(F59 F69
 5B  ------
 59
  :
 69
 70  MEDICAL     @ SUM(C72 C75                          @ SUM(F72 F75
 71  ------
  :
 75
 76  DOCTORS     @ SUM(C7B...CB2                        @ SUM(F7B FB2
 77  ------
 7B
  :
 B2
 B3  AUTO        @ SUM(CB5. .C90                        @ SUM(FB5. .F90
 B4  ------
 B5
  :
 91  INS.        @ SUM(C93...C97                        @ SUM(F93 .F97
 92  ------
 93
  :
 97
 9B  CONTRIB     @ SUM(C100. .C105                      @ SUM(F100. F105
 99  ------
100
  :
105
106  RDG.MTL     @ SUM(C108.. C116                      @ SUM(F10B .F116
107  ------
108
  :
116
117  LOANS       @ SUM(C119..C123                       @ SUM(F119...F123
11B  ------
119
  :
123
124  CLOTH'G     @ SUM(C126. .C134                      @ SUM(F126 .F134
125  ------
126
  :
134
135  RECREAT     @ SUM(C137.. C145                      @ SUM(F137 .F145
136  ------
137
  :
145
146  TAX/LEG     @ SUM(C14B.. C153                      @ SUM(F14B...F153
147  ------
14B
  :
153
154
155  OTHER       @ SUM(C157...C162                      @ SUM(F157 .F162
156  ------
157
  :
162
163  ============================================================================
164  TTL EXP     @ SUM(B21.. B162                       @ SUM(E21...E162
165  YTD          +B16*1                                 +B165+E164
```

Figure 2. Expense farmat.



	A	S	T	U
18	<EXPENSE	CURSOR	6-MONTH	
19	ITEM		TTL	DETAIL
20	----------------	----------------	----------------	--------
21	MORTGE		@SUM(B21...Q21	DETAILED TOTALS WERE DERIVED BY SUMMING ROWS C THROUGH R, E.G. BANK CH @SUM(C158...R158 FOR EACH ITEM
24	FOOD		@SUM(B24...Q24	
33	UTILITY		@SUM(B33...Q33	
41	HSHOLD		@SUM(B41...Q41	
51	RPR/MNT		@SUM(B51...Q51	
57	PERSNL		@SUM(B57...Q57	
70	MEDICAL		@SUM(B70...Q70	
76	DOCTORS		@SUM(B76...Q76	
83	AUTO		@SUM(B83...Q83	
91	INS		@SUM(B91...Q91	
98	CONTRIB		@SUM(B98...Q98	
106	RDG MTL		@SUM(B106 ..Q106	
117	LOANS		@SUM(B117...Q117	
124	CLOTH'G		@SUM(B124...Q124	
135	RECREAT		@SUM(B135...Q135	
146	TAX/LEG		@SUM(B146...Q146	
155	OTHER		@SUM(B155 ..Q155	
163	================	================	================	========
164	TTL EXP		@SUM(T21...T162	

Figure 3. Six-Month Summary Farmat

For use at the end of your six-month period, you can type the total and detail information as illustrated in Figure 3.

Extend Your Trip to a Year. It is possible to enter an entire year's transactions on your electronic worksheet. The examples in Figure 3 cover six months to ensure plenty of storage room. Storage room of 25K may seem like a lot, but it's surprising how much a person with a vivid imagination can consume. If your need is insatiable, however, you can always add a 16K memory card to your 48K Apple. The procedure followed in this example merely carries your six-month summary over to a second half-year worksheet and allows you to sail merrily along through year's end.

Now you are ready to put in your formulas. You can apply formulas similar to the examples. A couple of preliminary measures should prove helpful as well. Set your vertical titles in Column A, so that when you scroll from month to month, the titles will always be visible and, thus, ensure that you are making entries in the correct spaces. You probably won't want to set your horizontal titles for Income and Expense, since you'll be working first with one and then the other independently. Use the command /TV to set columnar titles.

The result of entering your formulas will make life forever a joy for you, although you may begrudge the task while you're involved. Just keep reminding yourself that your efforts now will eliminate the continuing need to add endless rows and columns to get monthly, year-to-date, and grand totals, because *VisiCalc* will now do all this for you automatically.

How It Works. Here's how some of the functions most useful in this program are executed. Others you will find in your manual and on your reference card as you discover more applications for other programs.

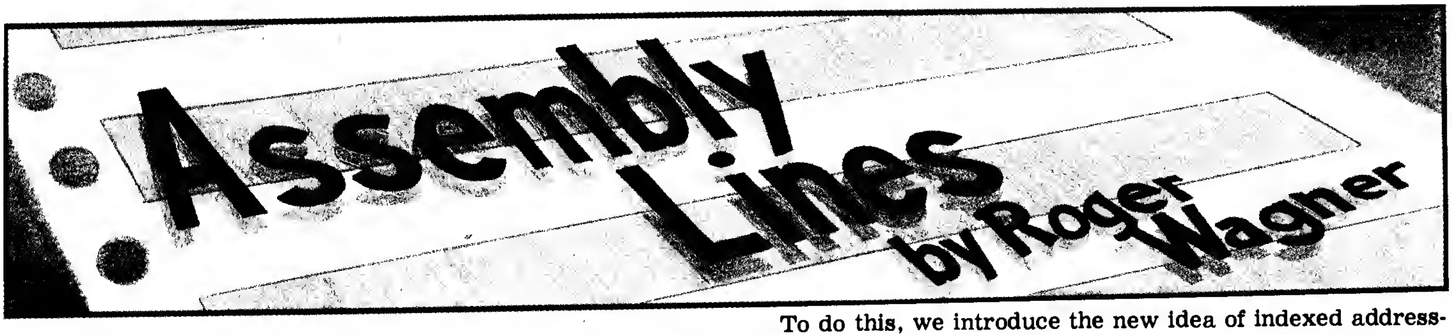
For summing:	'SUM(A1...A2
Split screen:	/W
Column width:	/gc
Print:	/P

You'll find all sorts of useful ways to use *VisiCalc* commands as you experiment with them. One tip for your checkbook program: narrowing your screen to five characters per column using integers will enable you to print a quarter of a year at a time, including the title column.

Whatever tricks you use, the end result will be less work for you in the long run, greater accuracy in your accounts, fewer headaches for you and your accountant, plus the ability to know where you stand at a glance regarding how you are earning and spending your money.

When tax time rolls around, instead of a thick book of handwritten figures, you can zip off a printout of your year's transactions, hand it over to your tax person, and relax.

If you have a VisiCalc application you'd like to share with Softalk*'s readers, write to Softalk Ventures, 10432 Burbank Boulevard, North Hollywood, CA 91601.*



Everyone's Guide to Assembly Language

Commands Covered So Far:

JMP	LDA	LDX	LDY	TAX
JSR	STA	STX	STY	TAY
RTS	INC	INX	INY	TXA
—	DEC	DEX	DEY	TYA
—	CMP	CPX	CPY	—
BEQ	BNE	BCC	BCS	—

This month we'll look at the various addressing modes used in machine language programming. This is a rather fundamental concept in programming and you may justifiably wonder why we have not covered it sooner. Well, as it happens, we have; I just didn't call it by name at the time. In the very first installment, we laid out the basic structure of sixty-four thousand individual memory locations. Over the last six months, we've worked most of our magic by simply manipulating the contents of those locations.

Flexibility in the ways in which you can address these locations is the key to even greater power in your own programs.

Consider this chart of the addressing modes available on the 6502:

ADDRESSING MODE	EXAMPLE	HEX BYTES
Immediate	LDA #$A0	A9 A0
Absolute	LDA $7FA	AD FA 07
Zero Page	LDA $80	A4 80
Implicit/Implied	TAY	A8
Relative	BCC $3360	90 0F
Indexed	LDA $200,X	BD 00 02
Indirect Indexed	LDA ($80,X)	A1 80
Indexed Indirect	LDA ($80),Y	B1 80

In looking at the examples, you should find all but the last three very familiar. We have used each of them in previous programs presented in this series.

The immediate mode was used to load a register with a specific value. In most assemblers, this is indicated by the use of the number sign (#) preceding the value to be loaded. This contrasts with the absolute mode in which the value is retrieved from a given memory location. In this mode, the exact address you're interested in is given. Zero page is just a variation on the absolute mode. The main difference is the number of bytes used in the coding. It takes three in the general case; in zero page, only two are required.

Implicit, or implied, is certainly the most compact instruction in that only one byte is used. The TAY command, transfer accumulator to the Y register, needs no additional address bytes because the source and destination of the data are implied by the very instruction itself.

Relative addressing is done in relation to where the first byte of the instruction itself is found. Although the example interprets it as a branch to a specific address, you'll notice that the actual hex code is merely a plus or minus displacement from the branch point. This too was covered in a prior issue.

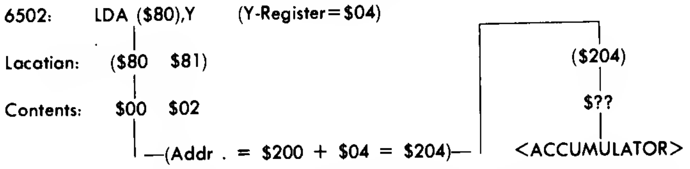
With this much, we can create quite a variety of programs, as is certainly evident in the programs being entered in our contest. (If you haven't sent your entry yet, don't procrastinate any longer—mail it today! Details appear in the March issue of *Softalk.*) The problem with these modes is that the programs are rather inflexible to data from the outside world, such as that in input routines, and in doing things like accessing tables and large blocks of data.

To do this, we introduce the new idea of indexed addressing. In the pure form, the contents of the X or Y register are added to the address given in the instruction to determine the final address. In the example given, if the X register holds a 0, the accumulator will be loaded with the contents of location $200. If, instead, the X register holds a 04, then location $204 will be accessed. The usefulness in accessing tables and the like should be obvious.

The problem that arises here occurs when you want to access a table that grows or shrinks dynamically as the data within it changes. Another problem occurs when the table grows larger than 256 bytes. Because the maximum offset possible using the X or Y register is 255, we would normally be out of luck.

The solution to this is to use the indirect indexed mode. This is really an elegant method. First, the 6502 goes to the given zero page location (the base address MUST be on zero page). In our example, it would go to $80 and $81 to get the low-order and high-order bytes of the address stored there. Then it adds the value of the Y register to that address.

INDIRECT INDEXED ADDRESSING


```
6502:      LDA ($80),Y     (Y-Register=$04)

Location:   ($80   $81)                         ($204)

Contents:    $00   $02                           $??

              —(Addr . = $200 + $04 = $204)—   <ACCUMULATOR>
```

Oftentimes, these two-byte zero-page address pairs are called pointers, and you will hear them referred to in dealing with various programs on the Apple. In fact, by looking at pages 140 to 141 of the Applesoft Reference Manual, you will observe quite a number of these byte pairs used by Applesoft to keep track of all sorts of continually changing things, like where the program is, the locations of strings and other variables, and many other nifty items.

If we wanted to simulate the LDA $200,X command with the indirect mode, we would first store a #$00 in $80 and a #$02 in $81—00 and 02 being the low-order and high-order bytes of the address $200. Then we'd use the command LDA ($80),Y.

Sometimes, X and Y Aren't Friendly. You may have noticed that I used the X register in one case and the Y register in the other. It turns out that the X and Y registers cannot always be used interchangeably. The difference shows up depending on which addressing mode and what actual command you are using (LDA, STX, and others). As it happens, indirect indexed addressing can only be done using the Y register. To know what is legal, you should make use of one of the many books on machine language programming or look at the chart on pages 121 to 126 of the newest Apple Reference Manual.

By now, you may have found that if you already know what the Apple-related books reference are trying to tell you, they're pretty easy to read. On the other hand, if you don't understand the topic, they're often of little help. Hopefully, this unfortunate catch-22 is bridged by articles such as Assembly Lines. The charts in the reference manual may appear rather confusing and vague at first. On further examination, though, I think you'll find they're just very concentrated in terms of information.

Referring to the entry for LDA on page 123, let's see what they're trying to tell us. In the first column is the mnemonic for the command, and below it is a short translation of the mnemonic. The next column gives a symbolic representations of the action that takes place when the command is used. The key to the symbols found here is listed on page 119. Usually, this column does little to enlighten; chances are the verbiage below the mnemonic will be of more help in explaining what happens. The third column is the one we're most interested in at this point. It shows which of the addressing modes are available for the command of interest. You'll notice that not all modes and index registers are available for each command. For example, for the LDA command, you can use either LDA $200,X or LDA $200,Y, but you can only use LDA ($80),Y.

The fourth column gives the usual syntax for using the command in most assemblers and the way in which the Apple disassembler will list the code to the screen when used. The hex code given is the byte that will be stored in memory to represent this command. The number following that in column six is how many bytes are used by the entire command. Compare the chart given earlier and verify for yourself that it agrees with the hex codes and number of bytes as listed in the reference manual. Just for fun, I purposely made an error in the commands covered so far as listed at the beginning of this article. Can you find it?

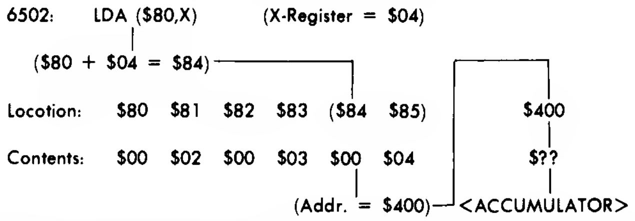
The last column indicates which flags in the status register are affected by using the given command. The order here is a bit different than the way they were listed in an earlier article, but you should still be able to find the Z flag (for zero result) and the carry flag. A check (√) indicates that the flag is affected by the command.

This chart is essentially your vocabulary list, and by now you'll notice that it contains quite a few of the commands we have covered. Look over the chart for the commands you know and see if the description matches your current idea of the command.

The last addressing mode, indexed indirect, is probably the most unusual. In this case, the contents of the X register (the Y register cannot be used for this mode) are added to the base address before going to get the contents. In a case similar to the other one, if the X register held 0, an LDA ($80,X) would go to $80 and $81 for the two-byte address and then load the accumulator with the contents of the indicated location. If, instead, the X register held a 04, the memory address would be determined by the contents of $84 and $85!

Usually, then, the X register is loaded with multiples of 2 to access a series of continuous pointers in zero page.

INDEXED INDIRECT ADDRESSING


6502: LDA ($80,X) (X-Register = $04)
($80 + $04 = $84)
Locotion: $80 $81 $82 $83 ($84 $85) $400
Contents: $00 $02 $00 $03 $00 $04 $??
(Addr. = $400)
<ACCUMULATOR>

Before we can put all this new information to work, we now need to answer one more question. How do you store just pure data within a program? All the commands we've covered so far are actual commands for the 6502. There is no data command as such. What is available though are the pseudo-ops, or assembler commands, of your particular assembler. These will vary from one assembler to another, so you'll have to consult your own manual to see how your assembler operates.

In general, the theory is to define a block of one or more bytes of data and then to skip over that block with a branch or jump instruction when executing your program. Usually, data can be entered either as hex bytes or as the ASCII characters you wish to use. In the second case, the assembler will automatically translate the ASCII characters into the proper hex numbers.

Most assemblers have a HEX command for directly entering the hex bytes of a data table. The *DOS Tool Kit* assembler is one exception; using it, line 20 should read, 20 DATA ASC DAPPLE. A sample program using the indexed address mode is given here:

```
 1  ************************
 2  * SAMPLE DATA PROGRAM *
 3  ************************
 4  *
 5     ORG $300
 6     OBJ $300
 7  *
 8  COUT EQU $FDED
 9  *
10  START LDX #$00
11  LOOP LDA DATA,X
12     JSR COUT
13     INX
14     CPX #$05
15     BCC LOOP
16     LDA #$8D
17     JSR COUT
18  EXIT RTS
19  *
20  DATA HEX C1D0D0CCC5
21  *
22  * DATA = 'APPLE'
```

When looked at in memory, it should appear like this:

```
0300-  A2 00     LDX  #$00
0302-  BD 0E 03  LDA  $030E,X
0305   20 ED FD  JSR  $FDED
0308-  E8        INX
0309-  E0 05     CPX  #$05
030B-  90 F5     BCC  $0302
030D-  60        RTS
030E-  C1 D0     CMP  ($D0,X)
0310-  D0 CC     BNE  $02DE
0312-  C5 00     CMP  $00
```

This program is an improved version of the one we did earlier to print the word APPLE on the screen. It uses the indexed address mode to scan through the data table to print the word APPLE. Notice that data tables may be wildly interpreted to the screen when disassembling. This is because the Apple has no way of knowing what part of the listing is data and tries to list it as a usual machine language program.

Basically, the idea of the program is to loop through, getting successive items from the data table using the offset of the X register. When the X register reaches 05 (the number of items in the table), we are finished printing. After printing, we terminate with a carriage return. Remember that in machine language we must usually do everything ourselves. This means we cannot assume an automatic carriage return at the end of a printed string.

Note that the hex values in the data table are the ASCII values for each letter plus $80. This sets the high bit of each number, which is what the Apple expects in order to have the letter printed out properly when using COUT.

The indirect addressing modes are used when you want to access in a very compact and efficient way. Let's consider the problem of clearing the screen, for instance. We want to put a space character in every memory location in the screen block ($400-$7FF). Here is one way of doing this:

```
 1  ***********************
 2  * SCREEN CLEAR PROG. 1*
 3  ***********************
 4  *
 5    OBJ $300
 6    ORG $300
 7  *
 8  PTR EQU $06
 9  *
10  ENTRA LDA #$04
11    STA PTR+1
12    LDY #$00
13    STY PTR
14  * STS PTR (6,7) TO $400
15  START LDA #$A0
16  LOOP STA (PTR),Y
17    INY
18    BNE LOOP
19  NXT INC PTR+1
20    LDA PTR+1
21    CMP #$08
22    BCC LOOP
23  EXIT RTS
```

Listed from the monitor, it should appear like this:

```
*300L

0300-   A9 04     LDA   #$04
0302-   85 07     STA   $07
0304-   A0 00     LDY   #$00
0306-   84 06     STY   $06
0308-   A9 A0     LDA   #$A0
030A-   91 06     STA   ($06),Y
030C-   C8        INY
030D-   D0 FB     BNE   $030A
030F-   E6 07     INC   $07
0311-   A5 07     LDA   $07
0313-   C9 08     CMP   #$08
0315-   90 F1     BCC   $0308
0317-   60        RTS
```

We start off by initializing locations $06 and $07 to hold the base address of $400, the first byte of the screen memory area. Then we enter a loop that runs the Y register from $00 to $FF. Since this is added to the base address in $06,07, we then store an $A0 (a space) in every location from $400 to $4FF. When Y is incremented from $FF, it goes back to $00, and this is detected by the BNE on line 18. At zero, it falls through and location $07 is incremented from $04 to $05, giving a new base address of $500. This whole process is repeated until location $07 reaches a value of $08 (corresponding to a base address of $800), at which point we return from the routine.

By changing the value of the #$A0 to some other character, we can clear the screen to any character we wish. In fact, you can get the value from the keyboard as we've done in earlier programs.

Here is a revised version:

```
***********************
* SCREEN CLEAR PROG. 1*
***********************
*
  OBJ $300
  ORG $300
*
PTR EQU $06
CHAR EQU $08
KYBD EQU $C000
STROBE EQU $C010
*
ENTRY LDA #$04
  STA PTR+1
  LDY #$00
  STY PTR
* SETS PTR (6,7) TO $400
READ LDA KYBD
  CMP #$80 ; KEYPRESS?
  BCC READ ; NO, THEN TRY AGAIN.
  STA STROBE; CLEAR KYBD STROBE.
  STA CHAR
CLEAR LDY #$00
  LDA CHAR
LOOP STA (PTR),Y
  INY
  BNE LOOP
NXT INC PTR+1
  LDA PTR+1
  CMP #$08
  BCC CLEAR
AGAIN JMP ENTRY
*
```

It should appear like this in listed form:

```
*300L

0300-   A9 04      LDA   #$04
0302-   85 07      STA   $07
0304-   A9 00      LDY   #$00
0306-   84 06      STY   $06
0308-   AD 00 C0   LDA   $C000
030B-   C9 80      CMP   #$80
030D-   90 F9      BCC   $0308
030F-   8D 10 C0   STA   $C010
3012-   85 08      STA   $08
0314-   A0 00      LDY   #$00
0316-   A5 08      LDA   $08
0318-   91 06      STA   ($06),Y
031A-   C8         INY
031B-   D0 FB      BNE   $0318
031D-   E6 07      INC   $07
031F-   A5 07      LDA   $07
0321-   C9 08      CMP   #$08
0323-   90 EF      BCC   $0314
0325-   4C 00 03   JMP   $0300
```

Enter this program and run from Basic with a CALL 768. Each key press will clear the screen to a different character. The screen should clear to the same character as the key you press, including space bar and special characters. In this program especially, I think you can see how fast machine language is. To clear the screen requires loading more than one thousand different locations with the given value. In Applesoft, this would be quite slow by comparison. I think you'll find that the screen will clear to different characters just as fast as you can type them.

An interesting variation on this is to enter the graphics mode by typing in GR before calling the routine. Then the screen will clear to various colors and different line patterns.

In this second variation on program #1, we've used the principles from the earlier issue of reading the keyboard until we get a value greater than $80, which means that a key has been pressed. This value is held temporarily in the variable CHAR so that it can be retrieved each time after incrementing PTR in the NXT section.

See what variations you can make on this, or try the hi-res screen ($2000 through $3FFF).

Next month we'll do some basic math operations, and look at some sound routines! ⊐■

Your Basic exec teom, from left ta right starting at the tap: Bill Gates ond Paul Allen, the founders of Microsoft; Vern Roburn, president of Microsoft Cansumer Products; Steve Ballmer, MCP's generol monager; Nick Roche, new notionol sales manager for MCP who signed on from Cammadore to assume responsibility far dealer and end user sales; Greg Tibbetts, monager of technicol support who is usually the vaice answering technical telephone queries; and Dottie Hall, monoger of communicotions

EXEC MICROSOFT

BY ALLAN TOMMERVIK

Tools, software with a difference, and unique hardware are the watchwords at Microsoft Consumer Products—the company that has brought you *Adventure, Typing Tutor*, the *SoftCard*, and a 16K RAM expansion board in its first twelve months of operation.

Adventure and *Typing Tutor* are regulars in *Softalk*'s Top Thirty and the *SoftCard* is well on its way to becoming the third most popular non-Apple hardware adjunct to the Apple, trailing D.C. Hayes's *Micromodem* and Mountain's Clock.

In terms of diversity of product—a fantasy adventure, an educational program, and a hardware addition—and of that product's marketplace acceptance, Microsoft's would be a successful record for any of the companies that preceded MCP into the personal computer business.

Want To Share Wealth of Progress. But for all of that success, nurtured from inception by MCP's president, Vern Raburn, the refrain that runs through Consumer Products is one of growth. And it's clear that a prime impetus of that desire is a frustration felt throughout both the division and its parent company that the development tools which have made the parent organization so successful are not being applied extensively outside the company in the microcomputer industry.

Microsoft was noted for its language development skills long before Consumer Products was started. Their methodology has always been to work within the framework of a larger system and download to the object system.

In the course of this method of development, the company wrote several software packages for their in-house use. This software, considered by founders Paul Allen and Bill Gates as their tools, provides MCP with a foundation of support for third-party authors that Raburn believes is unparalleled in the industry.

MCP is increasing its emphasis on reaching these outside authors as a means of better utilizing the tools they have developed and as a means of fueling their growth. Raburn believes that Microsoft's original success has had a tendency to detract from MCP's ability to attract outside authors.

Most Software Comes from Outside. "Microsoft has always had the reputation of doing everything in-house. Most of the independent authors don't realize that several of MCP's products are from outside sources and that we're actively seeking more such packages."

Bill Houlton photos

Adventure with an Expanding Apple

Adventure and *Typing Tutor*, MCP's best-selling products, were both written outside the company and their next product to be released for the Apple, *Olympic Decathlon,* is also an outside effort.

By no means is MCP getting into a bidding war with other software publishers for talent, however; and by no means is the company looking for mundane entries just to fill out their catalog.

Raburn offers support he believes to be unique to authors and looks for authors and products worthy of that support.

In little more than a year, MCP has earned a reputation in the marketplace as a name that means different programs of quality. Its success in that regard stems partially from breaking out all their considerations into marketing and publishing aspects.

Perfect Record Earned by Quality Products. From a marketing standpoint, MCP has an almost perfect record in assessing what products will be attractive to the end user. That success is enhanced by emphasizing documentation, design, and packaging as integral components of the product. Even after a year on the market, *Adventure* is one of the best packaged programs available.

Raburn has far from become complacent with his initial marketing successes. Instead, he's taking dynamic action to improve considerably the quality of MCP's sales and service.

First step in that procedure was arranging for time-sharing on a main frame to handle orders. With that system now in place, MCP can fill almost all orders within seventy-two hours.

More daring and innovative is Raburn's experiment of putting sales reps in the field to personally service retail accounts. The first of these reps is now working in the Pacific Northwest and, pending success of the program, Raburn hopes to have a nationwide sales force within eighteen months. His goal is to reach each retailer at least once in every three-week period.

Should that plan come to fruition, MCP would be the only software publisher with its own exclusive national field sales force. Hayden and Programma are the beneficiaries of Hayden's twenty-six-man field force, but they function under the aegis of Hayden Books and rep both books and software.

From the publishing end of the business, Microsoft may have unparalleled experience. They have licensed versions of their Basic to over one hundred fifty companies. Dealing with that many different systems has given the company in-depth understanding of the microprocessor that it can bring to an outside author. That experience, coupled with the software tools they have developed, should be of great value to authors during program development stages.

Raburn Refused To Sit Back. Raburn, the man running MCP's show, is almost ideally suited in background for the task. He is one of the few top-level execs—perhaps the only one—to have extensive experience at the retail and distribution levels as well as in publishing.

The experience was not all by design, but it's nevertheless stood him in good stead.

When Raburn graduated from Cal Poly, San Luis Obispo, he joined Minnesota Mining and Manufacturing's Graphics Division as a technician. Within nine months, he had been promoted to supervisor and looked to have a fine future with the firm.

But there were two bad chips on the mother board: First, he was told that his rapid ascension would have to come to a halt while he gained experience, and second, he was refused transfer into the Magnetic Media Division.

These problems fueled a previously latent desire to run his own store. With a former college roommate also located in southern California, he set out to form some enterprise upon which to hang a shingle. After several false starts in other industries, they succeeded in capturing an Orange County franchise from the Byte Shops.

It Wasn't His Thing, But He Profited. The retail business was a real education for Raburn. "We didn't know much about computers and chips, but the customers were wonderful. Those that had more knowledge than we were very helpful in teaching us what we didn't know. As we learned, those with less knowledge, but who felt a desire for a computer, appreciated our assistance, however inexpert. It was a crazy and exciting time. We were all like a big club, helping each other."

One of the things that Raburn learned, however, was that he wasn't cut out for retailing.

The Byte Shops' franchising operation was booming at this time. They had preceded Computerland by a year and were expanding at almost a geometrical rate. Raburn was brought in at the corporate level in the franchise and product marketing group.

There he learned from a different perspective about products, distribution, and retailing. Unfortunately, Byte Shops were growing in the face of adverse economic conditions. Money was tight, and as with almost every other technology-oriented business, Byte Shops had started undercapitalized. It was a fatal combination that would eventually call for contraction of corporate activities.

A GRT Noise in Software. Raburn left prior to the more massive layoffs to explore other new business opportunities. But even with the backing of a Bay Area venture capital firm, financing was not forthcoming.

At that time, GRT was making noise in the software marketing field. GRT was the multimillion-dollar cassette and tape marketing concern. They were feeling the pinch of competition from the major music companies and were looking to diversify. Software seemed to be a logical area, considering that the medium—cassette—was the same.

Raburn joined the product development and licensing

GOTO 50

Whatever Happens to Those Teenage Wonders?

Whatever happens to all the computer wunderkind?

You've seen them in practically every computer store and the totality of their knowledge about computers in general and the Apple in particular is awesome. They're the fourteen-to-seventeen-year-olds who can rattle off a driver program for your particular peripheral without consulting the reference manuals, can diagnose a chip problem in less time than it takes the Israelis to retaliate against the Palestinians, and can solve almost any programming problem you care to formulate.

First of the Bright Lights. There's a tendency to think of them as the first of a new breed and to speculate on their bright future in an increasingly computerized world. But they aren't the first generation of computerniks; there are precedents to predict the likely futures of these bright youngsters.

Two of the early wunderkind who can serve as models for the present generation are currently ensconced in Bellevue, Washington, where they direct the future of one of the most basic—pun intended—microcomputer companies in the country—Microsoft.

They're Bill Gates and Paul Allen, and their company touches nearly every personal computer owner because they authored the original microcomputer Basic language.

They estimate that more computer users—approximately eight hundred thousand—use one or more of their languages than use any other computer language in the world. Applesoft, a language most *Softalk* readers have a nodding acquaintance with, is an extension of Microsoft Basic, adapted to accommodate Apple's graphic capabilities and other nuances of the Apple architecture.

Back to the Basics. How Gates and Allen became proficient in computer science and the evolution of their interest into Microsoft can almost serve as a manual for those youngsters now following in their footsteps.

While still at Lakeside High School in the Seattle area, they had the opportunity to get extensive hands-on experience with the PDP-10. Computer Center Corp. was offering the time free of charge on the condition that the time be used to attempt to crash the system.

The pair had no trouble inducing the system to crash. In fact, a log book kept of the activities of all who used the system contained more than thirty different situations in which they brought the system down.

Rescuing a Giant. Although the experience was fun and Gates and Allen avidly pursued it, there clearly seemed to be no future for system-crashing talents in the real world. But, in the course of their activities, the pair had also gained extensive knowledge of the hardware, its capabilities, and the software of the system. This knowledge was to stand them in good stead shortly, when giant TRW was in dire need of PDP-10 experts.

TRW had the contract to develop and implement a computer system for real-time control of all power generated in the Columbia River basin, including the enormous Bonneville project. The PDP-10 was the chosen hardware.

A series of software bugs plagued the development and put implementation of the system so far behind schedule that TRW was to the point of having to pay forfeiture penalties for nondelivery. The company put out a worldwide call for PDP-10 experts. One of the resources they used in finding these experts was the log book of the Computer Center system that Gates and Allen had spent their afternoons trying to crash.

How They Spent Their Summer Vacation. Their names were on practically every page of the log and TRW made immediate inquiries into their availability. As it happened, Allen had just graduated from Lakeside High and Gates was between his sophomore and junior years there, so they spent the summer on the TRW project—Allen specializing in failure recovery and Gates in data storage.

The efforts of the two teenagers were vital to the eventual success of the project, but the endeavor also gave them an appreciation of system reliability requirements that proved most valuable in one of their later endeavors.

As Gates now wryly remarks, "The government had a reliability standard of 99.999 percent. Whoever heard of a requirement like that in the computer industry? When we got ready for the final test, we found that to meet that standard, we would only be allowed ten seconds of downtime.

"We held a meeting to determine how to allocate the ten seconds. The hardware people figured they would need several minutes of downtime, the software people were talking in terms of hours."

The final test went off without a hitch and the system was bought by the government.

A Hunger for Systems. Actually being paid for their efforts whetted the appetites of the pair to pursue other opportunities. They developed a system of reading and reporting the results of traffic tapes.

The tapes are generated by the counting machines most urban drivers are familiar with—the gunmetal gray box at the side of the road with a flexible black cable running from it across the street. Gates and Allen developed a superior reading system and successfully sold their services to many municipalities.

By this time Allen had matriculated to Washington State University and Gates was finishing high school.

Gates went off to Harvard and left Allen at WSU, supervis-

GOTO 51

MARKETALK

Reviews

Mastertype. By Bruce Zweig. Every once in a while, an idea seems so simple and natural when put into action that we fail to notice fully its ingenuity and value. This is a danger with Bruce Zweig's first foray into the software market, *Mastertype.*

Designed to teach typing and provide the massive drill needed to learn that skill, *Mastertype* is a lesson in how not to be boring. At the same time, since learning not to look at the keys is crucial to good typing, the author provides a visual distraction, in the form of a game.

Words to be typed appear on the screen in groups of four, positioned in the four compass positions. The user must strive to type these words in a rectangular box in the center. When you successfully type a word, it explodes—and a different word takes its place. The words aren't fond of exploding, however, so they begin firing missiles at your rectangle. You can save the day by shooting an oncoming missile, but the word that fired it must still be dealt with.

As you become more proficient and raise the speed of a lesson, the missiles also speed up. If you're working at an appropriate speed, you can't afford to look at your fingers lest you be detonated.

Lesson and speed are user choices; practice time on a built-in word processor is provided as an option; postlesson summary gives game score and typing analysis, as well as advice on your next steps. Incidentally, the speed designations are relative and are not based on words per minute.

Zweig knows well the fundamentals of skilled typing and provides for them. Seventeen lessons take the learner from single-key quasi-words to nine-letter words, then on to math problems, symbols, and numbers, in a format that begins with the home-row keys only, moves to concentrate on stretches, such as F-G or J-Y, but never loses sight of words. Except in the single-key and stretch lessons, there are no meaningless combinations of letters or symbols in the program.

Words can be terminated with a space bar or return, which provides equally well for typewriter typists who must get used to frequent returning and for computer word processor typists who rarely need return. If you're typing shows a bad habit pattern, program will interrupt the game to identify the error, then continue.

It is the sign of a good game when an officeful of people gather around one person playing the computer amid chuckles, cheers, and delight. It is very rare that such a reaction occurs with a nongame. But to experience this and hear cries of "I'm next!" being provoked by a disk to teach typing is a real tribute to the author.

Documentation is included but, except for the table of contents, unnecessary; full instructions are given (optionally) at the beginning of the program and commands are prompted clearly throughout. MCT

Mastertype, by Bruce Zweig. Applesoft, disk. $34.95.

Pascal Primer. *(Book review.)* By David Fox and Mitchell Waite. If you have wondered at the furor over Pascal, *Pascal Primer* is an excellent introduction. It presents a clear picture of the language that is rapidly becoming the *lingua franca* of the programming world.

Although the book was designed for people who have some knowledge of Basic, its clear, crisp style make it a good beginning book even for those who have never programmed. Authors David Fox and Mitchell Waite make learning to program in Pascal an easy and fun thing to do. The language is presented in a friendly and humorous way, and there are numerous source listings and examples of programming.

All programs used in the book were developed on an Apple, so the examples can be run without modification. This enables you to have direct experience with your Apple and enjoy the excitement of having programs work the first time they are compiled.

Because *Pascal Primer* is a beginning book, it doesn't cover the entire set of the language. It does explain the most often-used features.

The appendices contain valuable information for interfacing assembly language to Pascal. This is particularly helpful when using many of Apple's memory-mapped features.

Pascal Primer covers the necessary language information to start writing on your Apple Pascal programs that are instructive, easy, and fun to watch work. JV

Pascal Primer, by David Fox and Mitchell Waite. SAMS Publishing Corp. $16.95.

Space Eggs. By Nasir.

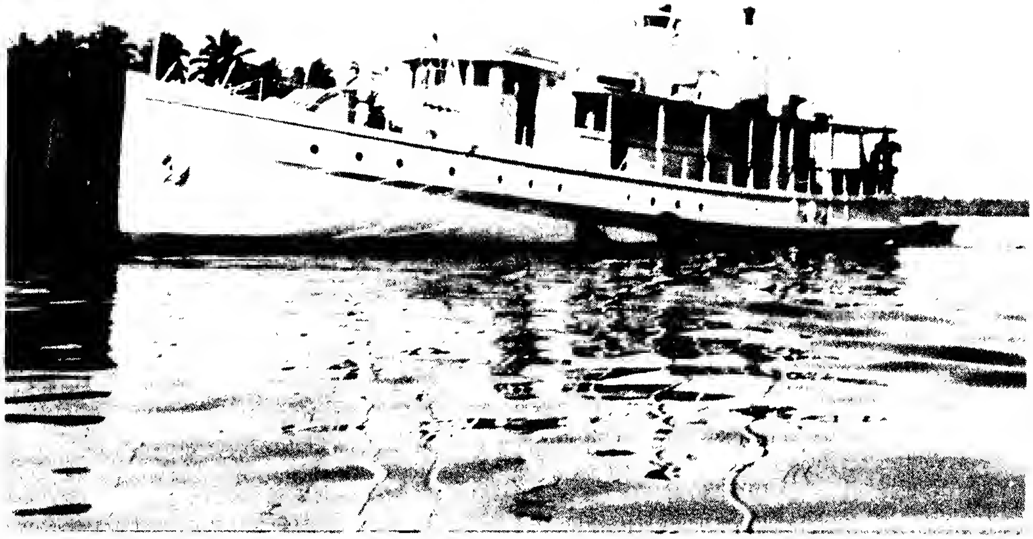
Phil Knopp
Sirius Software

Dear Phil:

Your message regarding your newest software program reached me this morning. Are you kidding? *Space Eggs?* Is this your Easter offering for the tiny tots?

I think the whole message must have been garbled. I can't believe you guys named something *Space Eggs* and I can't believe the description that was passed to me at all. Please confirm the following details:

GOTO 40



A Seafaring Apple
The Mazatlan Regatta

BY CRAIG STINSON

Try to imagine for a moment. If your Apple really did have a mind of its own, what would be its fantasy occupation? Would it be in show business? Sports? Run for president, perhaps?

How might it find the best of all worlds, where it could exercise its computing brainpower and its free spirit all at once?

An Apple belonging to Donald Gumpertz seems to have landed in such a micro utopia. For two weeks last November, Mr. Gumpertz's computer had the rare honor and privilege of monitoring the times and standings of some thirty ships in the Mazatlan Regatta.

Apples Away, My Boys. Snugly tied into one corner of the stately salon aboard the eighty-six-foot power yacht *Westward*, this happy 6502 got to travel along on the thousand-mile voyage as chief escort and communications post for the ships competing in the race and for the world at large.

Apple's crew: Serving as Apple's deck hands on its cruise to Mazatlan were Ann and Donald Gumpertz. The voyage was not all pleasure, Apple worked an hour a day monitoring the progress of entrants in the Mazatlan Regatta.

Not only was the ambience right, but the work was not all that tough, either. Every morning, the Apple had to be up at eight. As Gumpertz contacted each of the competing boats on the radio telephone, Ms. Gumpertz, his wife Ann, fed in their longitudes and latitudes at the keyboard.

The Apple's first job was to estimate the amount of time it would take each racer to reach the finish line, and second, by applying a previously determined handicap to each boat's time, to arrive at standings for the four different classes of yachts competing in the race.

The heavier computing had all been done ahead of time, by a landrooted mainframe. Fourteen hundred or so manipulations of some rather arcane data—involving water line length, sail area, shape of keel, weight, and other factors—had been required to arrive at the handicaps, called time allowances, for each competitor.

Just Ships That Pass in the Night. The reason for all this effort is that no two ships have exactly the same sailing characteristics, and, although all the racers would leave the starting line together, the test of a crew's skill and performance was to see not who would finish first, but who would have the best time as corrected by the handicap. Often, in such races, the difference in actual sailing time between the first ship to arrive and the winning entry would be on the order of six or eight hours.

So the Apple came aboard to convert raw data during the race into meaningful information. The whole procedure took perhaps a half hour in the morning and another half hour in the evening. The result of this labor at each session was a printout showing current position, estimated time of arrival at Mazatlan, corrected time of arrival, and standings for the four classes. When the Apple had finished its work, Gumpertz was back on the radio phone, giving the news to competitors and to the media.

Between times, the computer was at liberty to enjoy and reflect upon its surroundings of luxury and illustrious tradition.

An Illustrious Past. Prior to the 1980 Regatta, the job given to the Apple had been performed by a minicomputer owned by Don Douglas of Douglas Aircraft. The mini performed well enough but was unwilling to travel, and the business of reporting to and retrieving data from it twice a day was cumbersome and time-consuming. What was needed was a bit of mobility; hence the Apple was hired.

So that it wouldn't feel intimidated by following in the steps of a computer owned by Douglas Aircraft, the program used by the Apple was provided—and personally written—by Dr. Allen E. Puckett, chairman and chief executive officer of Hughes Aircraft. Dr. Puckett also was one of the competitors in the 1980 Mazatlan Regatta.

The *Westward* itself bears a history of exotic places and remarkable personalities. Built in 1924 by Ted Geary, one of the most prominent naval architects of his time, the yacht was in its early days a charter vessel, mainly plying the waters off Alaska. Gumpertz still carries on board an old sales brochure for the ship with a list of references that includes such names as Andrew Mellon, Irving Berlin, and Darryl Zanuck.

Around the World in 730 Days. Since 1967, when it came into the possession of its present owner, the *Westward* has been all over the globe. A map below decks charts a recently completed two-year around-the-world voyage in celebration of Don and Ann's marriage.

So the Apple is well cared for aboard the *Westward.* And even if plush comfort, the sea breeze, an easy job, and a surrounding of remarkable people were not enough, this computer also gets to share its environment with a host of other fabulous electronic gear—sonar to scan the depths, radar checking above the water, and other state-of-the-art navigational equipment.

Someone on the Transpac Committee recently contacted Gumpertz to discuss the possibility of using a microcomputer in a similar role on that race to Hawaii. So, if there are any out-of-work Apples out there that would like to send in a resume . . .

MARKETALK

Reviews

from 37

It's an Invader-type game with eggs floating about at the top of the screen. The player's mission is to crack the eggs, which are harmless, in order to get at the monsters contained inside. Now really, Phil, why can't we let sleeping eggs lie? Why on earth would anyone want to crack harmless eggs to be exposed to monsters—although I must concede that some of your monsters look even more harmless than your eggs.

I can understand the first phase, if I got the message correct. The eggs hatch into spiders. I guess that's scary enough, but what about the second stage! Lips? Nasir made *lips* into monsters? Is he anti-kissing? What happens if the lips *get you—mononucleosis?*

The way I grasp the scenario, if you escape the lips, you get space wolves. Now, that's more like it. Shooting down voracious wolves is a sport most gamesters can get into.

But I have to tell you, Phil, the message really got garbled at this point. I'm told that the most dangerous creatures, which you face in the fourth round, are fuzz balls! Either my info is wrong or Nasir's gone bonkers. Fuzz Balls?

I mean, it's bad enough that Tony Suzuki has turned innocent oranges into angels of gaming destruction, but *fuzz balls*? If I had a computer-operating cat, that might make sense, but we're presumably all human beings here, Phil, and I can't believe anyone will play a game where fuzz balls are the most menacing creatures.

Please confirm or deny the info I received asap as we might consider a review of the program if it's not as it was described to me. Thanks.

ART

Phil Knopp
Sirius Software

Dear Phil,

If it weren't for the problems *Space Eggs* caused at the office, this would be a thank-you note for sending us a copy of the program.

But the fact is that *Space Eggs* is one of the more beguiling creations for the Apple to come down the pike for a long time. Even members of our staff who aren't gamesters got hooked on this one. We knew we were in trouble when we heard groans and cries of "Dammit!" from the room where our printer was supposed to be cooking on manuscript.

We were in grave danger of missing our deadlines until we took the disk home.

This one rates right up there with *Galaxian* in the arcade genre. It's so captivating it could replace free love.

The most fiendish part of it all is trying to dock the first-stage weapon with the second stage. I think it's easier to organize a fund-raising dinner for Israel in Damascus than it is to dock those two ships.

And those fuzz balls are awful. Escaping their attack is almost impossible. The damn things even bounce!

But the sneakiest touch is the round that comes when you finally make it past the fuzz balls—then you can't predict what monsters will hatch from the eggs. Even the relatively innocuous spiders become more dangerous than Mount Saint Helens under those conditions.

I don't know if you can sell many of these until word of mouth spreads that fuzz balls really are fun to fight, but it should take off then. Cheers,

ART

Space Eggs, by Nasir. Sirius Software, Sacramento, CA. 48K, DOS 3.2 or 3.3. $29.95.

Multimusic. By Darragh Nagle. *Multimusic* is a novel approach to composing music for the Apple speaker. In addition to writing single-voice melodies with this program, you can write harmony in as many voices as you wish.

How, you may wonder, do you get harmony out of that humble instrument in the corner of your computer? The answer is that you don't, exactly; you get the illusion of harmony in somewhat the same way you get the illusion of continuity at the movies.

The speaker actually plays only one note at a time, but with *Multimusic* it can be made to strum the notes of a chord over and over again rapidly enough to create the effect of harmony. The sound may remind you somewhat of balalaikas or, perhaps, of the old tremolo style of theater organ music.

The program creates its own syntax for musical input. Some of the language will be familiar to those who already know musical notation; some of it will require a reorientation of thinking.

The notes still have their familiar alphabetic names, and their placement within the range of the system is straightforward: "C#3" produces a C-sharp in the third octave. Notes to be treated as a chord are entered sequentially, terminated by a period.

The readjustment has to do with the notation of rhythm. Two parameters are involved, the first one telling the system how many times to strum the chord, the second at what speed. If all your chords have the same number of notes, it's fairly simple to work with. But if you have a five-note chord followed by a four-note chord, for example, and you want both to have the same overall duration, then do some experimenting with the speed and reiteration parameters to get them to come out the same.

The system offers extensive, well-written instructions to get you through the learning process; they can be accessed at any point during input by calling "Help." There are also a number of other amenities, such as commands for making insertions or deletions, or appending to an already composed file. Compositions can also be transposed in either direction, in halftone increments.

There are plenty of challenges and rewards here for a creative imagination. The disk includes a number of examples to get you started, including a portion of the overture to *Messiah*, a rendition of "Happy Birthday," "My Bonnie Lies over the Ocean," and several others.

Playback of these and your own compositions is accompanied by an interesting bar-graph visual display through which you can follow the movement of the various harmonic voices. CS

Multimusic, by Darragh Nagle. Sonora Computing, Albuquerque, NM. 32K, Applesoft, disk. $25.

Micro*Painter. By Bob Bishop. *Micro*Painter* won't help you draw, and it won't put text on the screen. It won't enlarge or rotate or animate. But it will take any hi-res drawing or shape you've made some other way and allow you to paint it in as many as twenty-one colors with a few flicks of the paddle button.

The black lines of your drawing serve as barriers to the colors; each area surrounded by a black outline must be colored separately. So, where you have minute detail, you'll have to do a lot of positioning of the crosshair and button flicking. But if your sunny-day sky is all one light blue expanse, despite the tree reaching into it, move the crosshair to any spot on it and one flick will fill it with color.

This is only one facet of Micro*Painter. The many colors are made by combining dots of the six normal hi-res colors in patterns to create the illusion of different colors. But this tool doesn't stop there. Its microscope feature allows you to examine any spot in your painting close up. Each hi-res dot is enlarged to a half inch on an average-size television screen; the crosshair, which has no center, is correspondingly enlarged. In this mode, you can align on any one dot and change its color. This means you can shade from one color to another; add patches of deeper tone; or give warmth to a cold patch, such as blue, without the warm color showing.

No one who enjoys hi-res illustration should be without *Micro*Painter* or some program like it. The documentation is simple and fully illustrated; commands are uncomplicated and natural.

Children and hobbyists can enjoy this program as pure entertainment. That the package is equipped with real crayons and a tiny magnifying glass and is subtitled as an electronic coloring book is not misleading. The disk contains eight finished drawings ready to be colored; and few who finish them will be satisfied to stop there.

Combining this program with Nasir and Jerry Jewell's *E-Z Draw* or with Bob Clardy's *Higher Graphics* gives you a perfect combination of sketchpad and palette, future-style. MCT

*Micro*Painter*, by Bob Bishop. Datasoft Inc., Sepulveda, CA. Runs on any Apple with disk, color television or monitor. $34.95.

Hi-Res Cribbage. By Warren Schwader. If you're already a cribbage player, it'll take a few hands to get used to the computer-run game, but once you do, you'll enjoy a good competitor when there isn't a human to fill the post.

Hi-Res Cribbage assumes the strategy in the game to be confined to discarding to the crib and playing to peg, which longtime veterans of the game will recall as being accompanied by a series of two-line rhymes describing the status of each play. The computer doesn't know the rhymes, but it does know how to play its cards.

Counting your hand is done for you by the computer at lightning speed. Actually, not a lot is lost, since the one competitive aspect of that stage was the opponent's right to score any points in your hand that you overlooked, and this is impossible against the computer, which doesn't overlook anything.

The hi-res cribbage board is the elongated spiral kind; each player uses only one peg, instead of leapfrogging. Two more pegs keep track of games toward match. Well-done hi-res cards, which actually show red for the red suits, are dealt and you're asked to discard to the crib. Position of the crib reveals who begins the dealing; deal alternates thereafter.

Program offers slow or fast mode, but even in the slow mode the hands are counted and cleared away automatically if you take too long hitting return—a minor annoyance if you're frequently interrupted by kids or phones or bosses. In the fast mode, it seems as if you haven't a chance to count the hands at all, but a few passes at it reveals how capable the human mind is when challenged.

All points earned are indicated on the screen with the breakdown of their origin. An arrow indicates the leading player next to the number of points in the lead. At the end of each game, the match status is given. Skunking is possible, and the computer shows as much glee in recording your skunk, or double skunk, of it as it does in its skunking you.

Hi-Res Cribbage is accompanied by very strange, low-pitched sounds that are fun usually, but disastrous if you're trying to sneak in a game on your lunch hour. One refinement we'd like to see is the option to do without these.

However, cribbage is a good game, and the computer *Hi-Res Cribbage* plays a solid, challenging game. Cribbage has been overdue on the Apple, and its debut is a very honorable one. MCT

Hi-Res Cribbage by Warren Schwader. On-Line Systems, Coarsegold, CA. 48K, disk, DOS 3.2 or 3.3. $24.95.

The Star Gazer's Guide. By Kevin Bagley, revised by Robert C. Clardy. The main focus of this astronomy program is the constellations, making it ideal for young amateurs or casual sky watchers. Hi-res graphics distinguish by symbol between galaxies, various nebulae, and double stars, all of which are demonstrated with closeups of the kind of formation in question. Then you can choose to view the summer sky or the winter sky. No date is specified, and it is assumed that you will be using the program from the Northern Hemisphere.

After you've chosen a season, the screen will fill with the brighter stars visible that time of year; immediately following the stars to the screen are numbers showing the locations of forty constellations. Inputting any number will bring a closeup of that constellation.

Alternatively, you can call for a list of the constellations' names and numbers, then run them from there. Once you are looking at the stars that form the constellation, toggling the space bar will give you a description of the constellation: a little history, directions for locating the constellation in the sky, and a rundown of any unique astronomical phenomena in the configuration.

Pressing O (for outline) causes the constellation to be outlined in the modern method: geometrically. A second O replaces that outline with the graphical outline, which suggests the shape implied by the constellation's name. A third O, or an R, clears the outlines to reveal the stars alone again.

Written in a conversational manner and concentrating on those aspects of the skies that can be seen with the naked eye, *The Star Gazer's Guide* seems an ideal beginning for the person interested in stargazing more than in the science of astronomy.

Because it is also a fine starting program for novices who think they might enjoy the science, *The Star Gazer's Guide* is a fine complement to the more detailed and scientific *TellStar* (Marketalk Reviews, March 1981). You could comfortably begin here and move on to *TellStar* later.

If it is the constellations and their mythical forms and origins that interests you most, *The Star Gazer's Guide* is a pleasant way to learn.

The disk comes in DOS 3.2 but can be *Muffined*. The 3.2 disk must be entered with "Run Hello," as the program left no room for DOS. DOS will fit on the *Muffined* version. MCT

The Star Gazer's Guide, by Kevin Bagley with revisions by Robert C. Clardy. Documentation by Robert C. and Ann Clardy. Synergistic Software, Bellevue, WA. 48K Apple II, ROM Applesoft, disk drive. $30.

Marketalk Reviews IMPRESSIONS

The following programs were submitted to *Softalk* for potential review. We have looked at all of them; full reviews were rejected because either we lacked time or ability to examine them in sufficient detail, other products from the same publisher were more commanding of review, or they merely lost the space battle to other packages.

□ **Lords of Karma.** Avalon Hill (Baltimore, MD). Despite the large, colorful box, *Lords of Karma* is merely a text adventure game, albeit a fairly fun one. The price is right, but the cassette isn't. Avalon Hill, long a respected name in board games, seems overly penurious in its microcomputer division: games are available only on cassette so the company can produce a single product for three of the four major personal computers. Apple owners must advance tape past Tandy version to find Apple version. Saving to disk is possible, but still no simple procedure. Disk-saved version must be loaded from Integer Basic; then CALL-151 for monitor; then start with 800G. Otherwise program won't run properly. Once done, adventure is clearly the work of a different creative mind from those of Scott Adams or Roberta Williams and is enjoyable for its newness. Integer Basic, cassette. $20.

□ **Savage Island.** By Scott Adams, Adventure International (Casselberry, FL). First of a two-part adventure still has this reviewer going in circles between a beach, a volcano, and a lake, and being periodically eaten by a bear along the way. Looks like a winner for Scott Adams fans, with his special brand of illogical logic—if you fool around trying to do the im-

A Self-Taught Programmer Insures His Future

BY RICHARD KNUDSEN

Dick Clinchy is almost just another one of those folks who bought Apples on the theory they should be valuable in their business—without knowing just how or why.

Clinchy had a better clue than some first-time computer owners in that he had previously purchased a TRS-80, so he at least knew what he didn't want at the time he bought up to the Apple II.

County Creativity May Be Contagious. Perhaps Clinchy also had a head start in that he comes from fabled Bucks County, Pennsylvania, an area as heralded in literature as any similar rural area in America. Bucks County was the starting point of Levi Zendt, one of the protagonists in *Centennial.* James Michener, author of *Centennial,* now makes his home in Bucks County when in the country.

It may be something in the air that makes the area special and a hub of creative and interesting things. Even today, it seems in some ways to be a trendsetter. Recently, a dozen craftsmen formed a guild in Bucks County, returning their crafts to the organizational structure of medieval England, partially for the purpose of perpetuating their crafts.

But whether it's the Bucks County environs or just something about Dick Clinchy, the events that ensued from his purchase of the Apple give testimony to the fact that here's no ordinary Apple business user.

No Assurance of Insurance Software. Clinchy sells and brokers insurance for a living. In the rural environment of Springtown, Pennsylvania, he's able to conduct his business out of a wing added to the back of his home. That's where the Apple was installed and that's where the rest of this tale takes place.

It came as no surprise to Clinchy that there was no off-the-shelf software already in existence for insurance applications. He'd researched the subject enough to be prepared for that disappointment.

As insurance folk are wont to do, he'd done his homework and knew that with people like Mountain Hardware and D.C. Hayes supporting the Apple with high-class peripherals, and with applications like *VisiCalc* from Personal Software available, the Apple was the most versatile personal computer.

Specifically, it was the Hayes *Micromodem* and *VisiCalc* that sold Clinchy on the Apple. He felt it necessary to be in contact with larger time-sharing systems and the *Micromodem* represented the most reliable means for that task. The power of *VisiCalc* to aid an insurance agent—whose business is nearly all numbers—is almost self-evident.

What separates Clinchy from others in this fix is that he knew exactly what he needed to do to integrate the Apple into his business—he needed to learn programming.

Insurance Man Ignores the Odds. Getting conversant with programming is no easy task for any total layman, but it's even more difficult for those in rural areas such as Bucks County, which does not boast of an active users group and does not have several retailers offering programming classes and seminars, such as can be found in major metropolitan areas.

Unimpressed by the high odds against him, Clinchy set out to turn his Apple into an integral tool in his business.

Clinchy, along with literally thousands of other insurance salesmen in the United States, organizes himself with the aid of the Granum One Card system. This system enables him to determine most efficiently which of his clients should be contacted in a given month.

In the insurance business, it is de rigeuer to contact each client twice a year, once during the month of his birthday and once during the anniversary month of his insurance age change, which occurs six months before the client's birthday.

What this portends is that an insurance salesman contacts about 18 percent of his clientele each month—in the past using the Granum system of card filing to ascertain which clients were on tap for a specified month.

Begrudged the Birthdays. For an agent like Clinchy, with several years in the business and a mature practice, this entailed as much as a half day's work, sorting by hand through all the cards and noting which clients were due for contact and the reason for the contact.

Clinchy begrudged this time each month and decided to apply his Apple to the problem. In the fall of 1979, at the time of his purchase, about the most powerful data base program was *File Cabinet*, which was not adequate to the task.

So when he set out to do his own programming, it wasn't such mundane programming exercises as getting a line of text to print on the screen that occupied his efforts, it was developing a full-blown data base that would be responsive to the requirements on the Granum system.

Softalk photos

Most programming novices who are gainfully employed full time in fields other than playing with their Apple find very little time to increase their knowledge of the art. These owners usually are happy by the end of six months' ownership to be able to key in a program provided from a magazine or newsletter.

From Zilch to Data Base in Six Months. Making a full-fledged data base your first programming task is roughly comparable to tackling quantum physics theory with a junior high science background—a really quick study may make progress over the long haul, but the short-term outlook is bleak.

Not so with Mr. Clinchy, who had his data base almost perfected within a six-month period.

He depended on the Apple handbooks as well as a pair of magazine articles. One of the articles delineated the inner workings of *File Cabinet* and gave Clinchy a background on what the structure of a normal data base looked like. And a four-part series in *Creative Computing* entitled "How Not To Be Out of Sorts" provided him with further hints on how to structure his program usefully.

Equal parts of trial and error were also mixed into the recipe, with many long nights spent testing various methods to get around a programming problem.

Thus was born the *Life Insurance Client Management System*. The program will not make Bill Pasauer or Barney Stone jealous. Done in Applesoft, it's independent of the Apple's memory, because Clinchy realized that the number of clients would vary with each agent.

Not Speedy, But Very Time-Saving. Although the first draft of the program read names into the memory, sorted out the desired clients, and then printed that data out, now Clinchy has remodeled the data base to read in one name at a time off disk, evaluate that data file, and then read the next name.

This is nowhere as lightning fast as *DB Master* or *Data Factory* can sort through a comparable list of names.

But *LICMS* has cut hours off Clinchy's work time. The task that once took him several hours is now accomplished automatically by the Apple and a printer in a matter of minutes, while he goes about other business.

Even among those Apple owners extraordinary enough to accomplish a task as complicated as *LICMS* without prior background, Clinchy remains unusual.

Most would be willing to call it a day at that point, feeling they had already achieved much of what they bought the computer for. All this achievement did for Dick Clinchy was whet his appetite for harnessing the Apple to other tasks.

Metamorphosis of a Book . . . A significant portion of the insurance agent's time is taken up with calculations assessing the client's need in the perspective of his goals. Some of the more usual formats these calculations take are Capital Need Analysis or Financial Need Analysis, with the method used determined by the end goal for which the client is buying the insurance.

Methods for so calculating the requirements of the client have existed for some time, published in book form by Vernon Publishing Services, Inc., Vernon, Connecticut.

Clinchy went to Vernon with *LICMS* in hand and proposed that he be licensed to convert their published methods to software for the Apple-owning agent.

Vernon agreed and took the proposal one step further—why not sell a turnkey system, including hardware, to the agent without a computer?

To a Turnkey System. Thus was born LPA Tech Corporation, which will soon be marketing turnkey systems for insurance agents. The LPA recognizes the genesis of the company in Clinchy's brokerage business, which is called Life Plan Analysts.

In addition, Clinchy has harnessed *VisiCalc* to his task of calculating premiums, loan values over a period of time, and other necessary financial data about a proposed policy. He has developed a set of close to fifty templates for *VisiCalc* that will calculate almost any data an insurance agent would want.

The templates are sufficiently versatile that most insurance companies' tables can be inserted and the proper calculations made.

Clinchy also had found that as his agency prospered, he was paying less attention to after-sale reports. When he first went into business, he would prepare a lengthy report for a new client, explaining exactly what the ramifications of his purchase of insurance were. As his client base grew, he found the report getting out later and later and becoming shorter and shorter.

Now, the Apple can do the reporting for him in a timely and detailed fashion that better serves the needs of the new client in understanding the policy he just bought.

All of this has almost become a case of the tail wagging the dog. So Clinchy has retained Pierre L. DeCrouez, an experienced programmer who, among other things, will convert Clinchy's Applesoft programs into software compatible with the CP/M environment.

If You Want a Job Done, Give It to a Busy Person. If all this seems to have been the work of an insurance agent with time on his hands, let it be recognized that Clinchy had a successful agency prior to his Apple mania.

In addition to his base of life insurance clients, which ranges up and down the eastern seaboard and is not confined to Bucks County, Clinchy also manages a successful insurance brokerage business. In the brokerage end of the business, he specializes in placing impaired risk policies for agents whose home companies would not accept the policies.

Even though such a brokerage effort, on top of a thriving life insurance agency, would seem to be enough for one man, Clinchy is also active in his community and his industry.

Community activities include several functions with the Boy Scouts of America, an interest he shares with his son Sandy; he's a licensed emergency medical technician who takes his turn as a member of the volunteer Riegelsville Emergency Squad; he's active in the Upper Bucks County Chamber of Commerce and the local Rotary Club; and he sings with the professional Bel Canto group in Allentown, Pennsylvania. He also serves as a member of the Saint Lawrence University Resource Task Force, giving back to his alma mater some of that which it gave him.

In the industry, he's a member of the American Society of Chartered Life Underwriters, a director of the Estate Planning Council of the Lehigh Valley, a member of the National Association of Life Underwriters, and a member of the Life Underwriters Political Action Committee.

Clearly, here is a man who did not exactly have time to spare.

Time on the Fly. Does he regret the hours of slaving over the Apple? Not at all. That time is already being paid back to him in terms of increased productivity. He's now spending about 30 percent of his time on his computer projects; but the remaining 70 percent of his time in the insurance business is more profitable to him because of the time savings caused by the software he has developed.

In fact, Clinchy reports, the more he programs, the more he can accomplish for his insurance business. And that's a good thing, because, in his *spare* time, he's a fly fisherman who ties his own flies and helps his wife Gerry train their golden retrievers for field trials.

Adventure with an Expanding Apple

from 34

team, which at that time was working closely with Softape and Microsoft, among others.

Under the imprimatur of G-2, the software division immediately started showing a hefty profit. But that profit could not offset the staggering losses to which the parent company was being subjected.

Microsoft's *Level Three Basic* was G-2's leading product, but it, along with about a half-dozen other products, got frozen when the company ran out of money for more manuals, packaging, and tapes.

By this time, Bill Gates had become convinced that software was the key to the microcomputer market. He felt that the software provided the solutions for end users and sold systems. With that conviction, he was determined that Microsoft would compete in the consumer market.

An Opening at Level Three. The failure of GRT to provide the vehicle for that competition left Gates undaunted. He approached Raburn to take the helm of the new Consumer Products division he was contemplating. The position seemed to combine requirements for all the experience he had gathered, so Raburn immediately accepted.

MCP opened in December of 1979 with *Level Three Basic* as its only product. *Adventure, Typing Tutor*, and *Editor-Assembler Plus* were added early in 1980 to flesh out the product line.

The overview of the industry shared by Microsoft's executive team provides a context for their next product moves. And some of the company's past actions, put into that context, ratify their foresight.

As explained by Bill Gates, president of Microsoft, the company sees a shifting emphasis to software in the microcomputer industry.

"Software is really a separate industry. The hardware manufacturers can't supply all the applications software. In the past, software has been the slave of hardware. Now the cost of the computer is as close to zero as it can be and the focus is changing to the software."

Microsoft's own actions cast the shadow of self-fulfilling prophecy on its president's views.

Faced with the task of converting all of their 8080-based software to the Apple's 6502 environment, the company chose instead to develop the *SoftCard*, which makes the conversion within the Apple. It was a clear case of the software dictating a hardware solution, rather than the traditional problem-solving route of hardware dictating a software solution.

Hard Solutions to Soft Problems. One of the essential advantages of the *SoftCard* is bringing to the Apple the programs originally designed to run in the CP/M environment. Typically designed for systems with larger memory than the Apple, many of the most powerful CP/M programs remained unavailable to Apple owners.

Again, Microsoft came up with a hardware solution to a software problem—introducing the *RAMCard*, which provides the Apple with an additional 16K of RAM and making many more of the CP/M programs available to Apple owners.

Raburn takes a sheepish attitude toward the *RAMCard* for two reasons: the card is so basic that it fails to meet his requirement that MCP put out product unique to the marketplace. As he puts it, "We never would have put out the *RAMCard* if there had been someone else manufacturing a competing product at the time. But the memory extension was necessary to support the *SoftCard*, so we went with it."

The second cause of Raburn's embarrassment about the *RAMCard* is that MCP drastically underestimated the market for the product. Although MCP has a policy of being scrupulous about not hyping a product before they're ready to deliver it, they've found themselves considerably back-ordered on the *RAMCard*.

But the emphasis at MCP is not on hardware, it's software. And their emphasis to outside authors is on the transportable tools they can provide to assist the author.

To increase MCP's visibility, Raburn raided his parent company for Steve Ballmer, Stanford-educated former assistant to Gates, who is now general manager. Ballmer expands the tool idiom that suffuses MCP by pointing out that the individual author "provides the creative genius of the idea. Our tools will help him most easily realize that idea."

Synergy Makes Magic. Ballmer's goal is to get "lots of application software written using our tools."

Ballmer sees the magic of synergism at work at Microsoft. "Paul Allen's new products research feeds both the OEM and MCP arms of the company. The OEM division feeds tools to both the New Products group and MCP, as well as language products for marketing by MCP. And MCP provides the conduit through which the end user benefits."

When product finally hits the market, responsibility for its promotion falls to Dottie Hall, one of the first management people brought in by Raburn.

She brought to the company a breadth of understanding of the microcomputer industry gained through a stint of employment with *Computer Retailing*, one of the industry's trade magazines. She's been in charge of executing the advertising and promotions campaigns for each of the products, as well as displaying the company's products at various trade shows and expositions.

Microsoft Computer Products now supports its product line with a staff of fifteen. Hiring of several additional employees is already planned and scheduled. But the addition of personnel takes secondary importance to making their software tools available to authors for applications software development.

In Vern Raburn's view, "1980 was a year of building for MCP." Now is the time for the company to capitalize on its foundation.

Whatever Happens to Those Teenage Wonders?

from 34

ing a declining business. Most state governments had traffic tape reading devices and were finding that they had excess reading capacity, so they started offering the service free to municipalities. Not even a couple of the original computer wunderkind were able to figure out how to compete at that price and make a profit, so they were open to other possibilities.

Opportunity came in the form of summer employment offered by Honeywell. Gates was hired between terms at Harvard and recommended Allen when Honeywell sought additional help. Allen moved across the country and the two were reunited.

Taming the Wild 8080. It was shortly after this, in early 1975, that a story in *Popular Electronics* announced the Altair microcomputer, first of its breed. Allen had already proposed developing a Basic language for the 8008 microprocessor, an idea Gates had vetoed. But the advent of the 8080 microprocessor brought a meeting of minds that here was a chip worthy of their efforts. Because the Altair's basic component was the 8080, they called Ed Roberts, the Albuquerque-based developer of the computer, and offered to develop Basic.

Roberts told them that several other people had a head start on the same idea, but encouraged them to present their version of Basic if it could be ready soon. Allen and Gates promised one-month delivery and went to work. Allen spent two weeks designing an Altair simulator and an assembler. Gates spent that time generating design notes. Working from a larger system, it took them an additional four weeks to perfect their Basic.

They reported their success to Roberts who invited them to Albuquerque to show the language. It was not until Allen was on the plane that he realized the program lacked a boot-strap loader, so he spent the travel time devising one.

Improving on Perfection. Now came the big payoff on their time spent with TRW, working with those impossible reliability requirements. Although neither Gates nor Allen had ever seen an Altair micro, the program ran perfectly the first time it was loaded.

Gates was still at Harvard and Allen was at Honeywell, but Allen barraged Roberts with so many ideas on improvements to their Basic that Roberts finally invited Allen to work on them full-time as director of software development for Altair.

The agreement included Roberts using the company's best efforts to promote the language development programs that Gates and Allen had proved so proficient at producing. That agreement remained operative until Pertec bought out the original company. It didn't take Pertec long to realize that the beneficiaries of these language development activities were Pertec's mini and micro rivals. At that point, best effort nearly became synonymous with no effort.

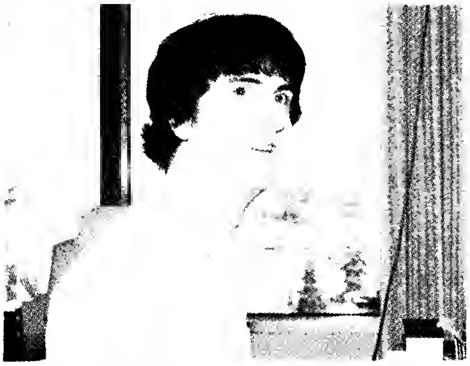
Gates and Allen finally extracted the most recent version of their Basic—now commonly called Microsoft Basic—from Pertec and formed Microsoft. Finding no further need to be located in Albuquerque, the pair moved their operation back to the Pacific Northwest.

Since that time, the company has provided language development services for almost every new small system—including some that never got past the prototype stage. Microsoft has expanded until it now has more than sixty employees, functioning in three distinct areas. Gates heads up the OEM division, which specializes in developing languages for original equipment manufacturers. Allen leads a task force researching new products for the company.

A Bull Market. And Vern Raburn serves as president of Microsoft Consumer Products, the division that markets the software and hardware available from your local retailer. MCP is barely a year old, yet Gates already foresees the day in the near future when it will pass the rest of the company in sales.

That MCP shows such potential within a company that now exceeds eight million dollars a year in sales is a tribute to Raburn's stewardship. That the company has established its languages as the standard in the microcomputer industry is a tribute to Gates and Allen, whose successes provide a sterling example to the current generation of computer wunderkind. It's that basic.

One of the New Wunderkind



Softalk photo

Neil Konzen

Microsoft is not only the home of two of the earliest computer wunderkind, it's now serving as a haven for some of the current generation. Neil Konzen has yet to reach his majority, yet he wrote the *Program Line Editor*, published by Synergistic Software and considered a must in every serious programmer's library, and he was chief author of the software for the *SoftCard*. The opportunity to work on the *SoftCard* took precedence in his eyes over continuing at the University of Washington. He's currently working at cramming Basic into a 4K hand-held microcomputer.

MARKETALK News

□ *Trend-Spotter,* a business forecasting and graphics package from **Software Resources Inc.** (Cambridge, MA), transfers data into hi-res graphic tables, making it easier to locate business trends. Color graphics enhance bar graphs, scatter graphs, line graphs, area graphs, or side-by-side bar graph forms, or any combination thereof. Package does statistical and mathematical computations, prints tabular and graphic data, calculates trend lines, and reads and generates *VisiCalc* compatible files. Statistical functions include compound inflation adjustment, rate of change determination, short-term or long-term trend lines, moving averages, and arithmetic operations. 48K, Applesoft Basic. $275.

□ **Corvus Systems** (San Jose, CA) presents five-megabyte add-on Winchester disk systems compatible with Apples. A compromise between the one-megabyte and ten-megabyte drives for multiuser systems, the five-megabyte drive for the Apple is designated the *5AP.* Systems include drive, Z80-based controller card, interface card for 6502-based software, and complete power supply. Drive has 5.8 megabytes formatted and 6.9 megabytes unformatted data capacities, minimum seek time of ten milliseconds, average seek time of fifty milliseconds. $3,750.

□ **Southwestern Data Systems'** (Santee, CA) CP/M based communications package, the *Z-Term,* permits the Apple to communicate with most computers with dial-up access. Regularly used systems can be put into a directory for auto-dialing, data files can be uploaded and downloaded between terminals, and keyboard macros (single keystrokes or control characters) allow user to define strings for output with simple keystrokes for fast log-ins to other systems or issuing commands in the system. The *Z-Term* can configure the Apple to emulate a different terminal, or, with the Apple as a translator, can define one terminal as another. Optional 16K RAM or language card increases internal buffer size. Requires Microsoft's *SoftCard.* $79.95.

□ A package for Apple farmers of any crops comes from **Rural America Enterprises** (Marshall, MN). *Farm Ledger* maintains financial records, generates balance sheets, income statements, balances and sales reports. Monthly or quarterly totals permitted as user requires. Package includes farm account chart, allows for custom agricultural charts and company statements. Company will customize *Farm Ledger* to your applications upon request. 48K, Applesoft, two disk drives, 132-column printer. $395.

□ **Computer Case Company** (Columbus, OH) has come to the aid of Apple III owners with a heavy duty portable case that can store the III, two additional disk drives, cables, and additional supplies and papers. Case is designed to protect computer from damage with padding and hard luggage frame and sides. Remove top, and you can leave Apple III in case for use. Apple may be secured within case to prevent tampering without having to disconnect cables. $139.

□ Several new games are being released from Creative Computing's (Morristown, NJ) **Sensational Software** line of games. *Story Time* is a cornucopia of illustrated tales in which children can include themselves, their friends, and their personal

crocomputer Products (San Jose, CA) increases the Apple's ability to interface with a large number of peripherals and terminals, and simultaneously with serial and parallel devices under Pascal. The serial interface has eight standard baud rates from 110 to 9600, and several nonstandard rates, all rotary switch selectable or via external input. Four additional interrupt and handshaking lines included in the two bidirectional eight-bit parallel ports, which have programmable and software-controlled interface configurations. $225; kit form, $175.

□ *The Community Computerist's Directory* is planned for twice-a-year publication in January and July by **Alternet** (Santa Rosa, CA). Publication consists of listings (500-character limit per) of computer uses by companies, groups, and individuals, providing reader with a single reference source for information, services, and products. July issue is planned as a *Yellow Pages* type directory, providing for detailed descriptions and information, supported by display ads; additional listings present user groups, bulletin boards, publications, software houses, retailers, and clubs. Noncommercial listings cost $5, commercial listings $10, two-by-three inch display ads $20. Catalog price is $4 per issue; January issue available only by mail order from The Community Computerist's Directory, Box 405, Forestville, CA 95436.

□ Another catalog, the *Software Vendor Directory*, is available from **Micro-Serve**. The fourth edition of this directory of software companies in the micro industry features 4,195 products, 80 hardware and 200 software categories, and 1,001 vendors. Directory with two updates costs $100; separate directory is $57.95, with individual updates at $25. Disk version, at $78, includes an information retrieval program from **Island Cybernetics** (Port Aransas, TX). For directory, write Micro-Serve, 250 Cedar Hill Avenue, Nyack, NY 10960. Master Charge, Visa, COD orders accepted. Mail-order only; no phone orders accepted.

□ *Professional Time and Billing* from **Software Technology for Computers** (Belmont, MA) maintains billing data for as many as three hundred clients. Program stores a maximum of sixteen hundred charges per current billing period, rate or employee files, and client files. Charges and payments can be entered and accumulated for clients on file. User can get reports on time and expenses, client activities, zero balances, and grand total statements. Compatible with Language System, 48K, Applesoft ROM, two disk drives. $325. Software Technology's *Coloring Board* draws pictures in hi-res using paddles. User can save shapes for use in personal programs and can utilize both hi-res pages for animation in six hi-res colors. 48K. $60. Same company's *Language Reading Development* offers a complete course in reading, accompanied by manual, exercise worksheets, full graphics, and sound. Teacher can add own exercises to program. 48K. $250. Latter two packages are distributed by Bell and Howell's educational arm, **UNICOM** (Providence, RI).

□ The video disk can be coupled with the Apple using the new *Interactive Computer Video System* from **Symtec** (Detroit, MI). System can receive video signals from an outside source, process them, superimpose text or graphics from the Apple, and then send out the composite video signal. Peripherals include Discovision Associates' 7820 video disk player, color monitor, and Symtec's light pen, video disk control interface, and video data processor. Light pen, allowing user to work with video images from Apple or video disk, returns the Apple to an X and Y coordinate value system for drawing onscreen. The video data processor, being independent from the Apple, synchonizes itself to the incoming video signal from video disk, tape, or camera. Processor has 16K of RAM with full color operation. $6,900 including system software.

□ A medieval adventure, *The Crown of Arthain*, challenges one or two players taking on the identities of the dying king of Arthain's two sons, princes Berthain and Merthain to penetrate monster-filled copses, ford streams, and traverse the mazelike underground mountain hall to achieve entry to the king's quarters and take possession of the crown. Dan and Marilyn Meller's game from **Micro Lab** (Highland Park, IL) has hi-res color, twenty levels of skill, and realistic animation. Even with advanced skill in swordsmanship, you will need the magic aids hidden along the way to keep you alive. 48K, Applesoft ROM. $35. Micro Lab also presents a comprehensive education package by Scot Kamins called *The Learning System*. Emphasizing teacher-student interaction, package runs instruction and test modes upon input of text, yet quizzes can be set up without reference to text. Instruction mode gives student two or three chances to answer questions; test mode permits only one chance then charts percentage of correct and incorrect answers and number of times required to arrive at correct response. Results can be saved for teacher's future use. Many evaluation methods featured, including comparing student performances on same exam, teacher performance, score averaging and estimating and class curves estimations. Package includes master, master back-up, and two player disks. 48K, Applesoft ROM. $150 introductory. A very busy Micro Lab offers as well the *Invoice Factory*, which stores and retrieves invoices; produces customer, product, or receivable reports; and creates annual bar graph analyses. One disk can handle approximately three hundred accounts. User can note shipping and handling methods and terms of payment; features UPS zone fee table. Compatible with company's *Data Factory* and *Mini Factory* packages. 48K, Applesoft ROM. $100 introductory.

□ A graphing tool from **Muse** (Baltimore, MD) called *Data Plot* transposes numerical information into graphic forms from single and multiple lines to bar graphs, pie charts, and scatter diagrams. Features include automatic and manual labelling and scaling. Records print out on Trendcom or Silentype printers, 48K, Applesoft ROM. $59.95.

□ A connection to the analog side of the computer world is available from **Street Electronics Corp.** (Anaheim, CA). Their *Apple Eight Channel A/D Interface* translates incoming analog signals to digital signals for the Apple to read. It samples eight analog quantities from zero to five volts in less than one millisecond or one channel at a sample rate of more than

10kHz. Board has high impedance input buffers and calibrated reference and can interrupt at end of a conversion. $99.50.

□ The *IBMS*, or *Interactive Business Management System*, is **Programma International's** (Burbank, CA) contribution to the business management field. Three years in the making, *IBMS* is a turn-key system with nine interactive modules; a data entry in one area updates related data in all other areas. Menu includes payroll, general ledger, mailing labels, accounts receivable, perpetual inventory, and more. Introductory price of $1,495. *Electric Template* from Programma aids the technical illustrator in drafting. Predefined or user generated template shapes follow Apple's shape table format; with game paddles or joystick, user can draw lines, in either black or white, with positioned cursor on hi-res screen. Besides template shapes for chemical, engineering, or electrical purposes, *Electric Template* provides a variety of geometric shapes and character shapes from normal to five times normal size. 48K, Applesoft ROM or Language Card, DOS 3.3 or 3.2, paddles or joystick. $49.95.

□ **ALF Products** (Denver, CO) offers a disk copying service for fifty or more disks. Cost ranges from $2.60 each for fifty copies to $2.10 each for five thousand copies, all on Memorex disks. Supplying your own disks cuts cost: $0.60 apiece for fifty to $0.20 apiece for five thousand. A setup charge for thirteen-sector or sixteen-sector disks is $10; charge is higher for special formats. Service to make disks copy-resistant is available from $25. Masters are kept on file for immediate reordering.

□ Hardware that assists user in the scientific laboratory comes from **Interactive Microware** (State College, PA). *APPLAB* collects or controls data from most scientific instruments, from pH meters to spectrophotometers. Board features a 32-bit real-time clock that displays times in hours, minutes, and seconds and times events down to 0.1 second. Two 16-bit timers may be configured as pulse counter or generator, interval timer, shift register, or short wave frequency generator. Included with each *APPLAB* card is *QUICKI/O*, a software program that facilitates the writing of Basic programs that interact with *APPLAB* controls. Fortified with extensive documentation, the *APPLAB* system may be expanded to three cards. Package includes card with disk, three cables, self-test adapter board, diagnostic software, and two manuals. $495.

□ *Algebra I* from **Edu-Ware** (Canoga Park, CA) helps the student in understanding and using beginning algebra. Instruction is step-by-step and comprehensive; features hi-res color graphics, upper and lower case text, and flow-charted information maps to chart student's progress. *Algebra I* is the first in a series of five instructional mathematics programs; mastery of one program leads to the next, more difficult one. 48K Applesoft, DOS 3.3. $39.95.

□ *Uni-Text*, a lower-case chip from **Dockside Computing** (Westlake Village, CA), provides screen display of all ninety-six ASCII printable characters, complete with true descenders—the only adapter on the market with this capability. For Basic and Pascal users, diskette software is provided; Pascal users should also note that proper inverse cursor is displayed with lower-case characters. *Uni-Text* operates with all popular word processors for the Apple. Chip based on EPROM device allowing for character set modification for custom applications. It's compatible with all software that can interact with every other lower-case adapter. $79.95.

□ Ken Williams and Jay Sullivan have jumped from the football gridiron to the soccer field with *Soccer* from **On-Line Systems** (Coarsegold, CA). With the world's most popular spectator sport finally making its mark on the American sports scene, it's only natural that it should make a mark on the Apple game scene. Realistically animated players battle on a hi-res screen. Game boasts real-time clock, hi-res scoreboard, three skill levels, and Williams and Sullivan's best sound effects yet. Every arcane rule of the game is accounted for; players have choice of playing by outdoor or indoor rules. 48K. $29.95.

TRADETALK

□ When Ringling Brothers Barnum and Bailey Circus gave up the last big top for permanent structures like Manhattan's Madison Square Garden, many regretted the passing of an era. Few events were more colorful than the circus train's arrival, the subsequent parade of exotic animals and circus performers to a barren field, and the metamorphosis of that field into the colorful lively atmosphere of the big top. It was sad to think it would soon be gone forever.

It wasn't. The modern-day version isn't a real circus, however; although the magic it presents is far more amazing than circus feats.

Instead of a train, there's a convoy of giant vans; instead of a tent, the big top is a modular, circular building eighty-five feet in diameter. Instead of three rings, the building houses six wedge-shaped theaters, like slices of an apple pie. The star is neither trapeze artist, wild animal trainer, nor clown: it's a computer.

If you live in Dallas, Chicago, New York City, or Los Angeles, you can watch the caravan pull into town—three forty-five-foot moving vans brightly painted in Apple's rainbow of colors and announcing on their sides, "Apple Computer Presents Apple Computer Expo—The Greatest Show on Apples!" You can watch Apple raise its version of the big top and attend the magic show later—the Apple Expo.

Aimed primarily at the Apple dealer, the Expo offers one day in each city for the public. There'll be ten-minute presentations on applications areas and a multimedia show giving an overview of the computer industry.

Accompanying Apple will be a number of software and peripheral manufacturers. Each will personalize and exhibit in an Apple-provided booth, ensuring an overall atmosphere of color 'and quality.

The public portion of the expo is the third day of each city's stand, from one to nine p.m. Admission is $10 per person at the door, but Apple is supplying Apple dealers within one hundred miles' radius of each city with as many free tickets as they can give away—so make sure you pick yours up at your dealer ahead of time.

Here's a quick rundown of the dealer-only portion of the expo. Day one: product seminars, hands-on and how-to seminars, talk by Steve Jobs or Mike Markkula; six seminars per hour. Day two: trade show including exhibits from Apple and from the accompanying peripheral and software manufacturers. Day four: Opportunities Day—by invitation only. This day is designed especially for the thousands of people who have shown interest in becoming Apple dealers. Highlights include seminars on the industry, Apple, and how Apple supports its dealers.

Apple Expo ends its Dallas run April 1. It will appear in New York April 5 through 8, in Chicago April 12 through 15, and in Los Angeles April 26 through 29. Check Dick Cavett's Expo ads in your city's newspapers for the precise location in which the Apple pie will be set up.

The train and big top of the circus are gone. But the caravan and modular pie of the Apple computer are here.

□ **Lou Long**, director of operations and advertising for **Stoneware** (San Raphael, CA), announces that the company's move to 50 Belvedere St., still in San Raphael, has tripled its office space and enabled a staff increase from three to fourteen full-time or part-time employees. Stoneware is now concerting even greater effort toward product support.

□ Before and after the 1981 national convention of the **Association for Education Communications and Technology** in Philadelphia, **Nova University** (Fort Lauderdale, FL) will be con-

TRADETALK

ducting eight microcomputer-in-education workshops covering computer literacy, instructional uses, and curriculum design. Registration fee for the workshops (#26-33) on April 5 and 10 is $40. Apply at workshop site, Philadelphia Civic Center, Pennsylvania Hall.

□ **Warren Schwader**, author of *Cribbage*, apparently was so taken with his game's publisher, **On-Line Systems**, that he pulled up stakes in his home of Janesville, Wisconsin, and moved to Coarsegold, California, to join On-Line's rapidly growing staff. **John Williams**, brother of On-Line president **Ken Williams**, has come on board as advertising manager. **Eric Griswold** is now assisting On-Line on copy protection, and the general staff has expanded by three people.

□ **Avant-Garde** has changed its address to Post Office Box 30160, still in Eugene, Oregon, according to **Mary Carol Smith**, one of Avant-Garde's founders.

□ Is Broderbund becoming Sisterbund? **Cathy Carlston**, sister of **Broderbund Software's** (Eugene, OR) founders **Gary** and **Doug Carlston**, joined the firm this month as director of advertising and public relations. Cathy was previously director of Christmas store operations for the Lord and Taylor chain of department stores, operating out of Manhattan.

□ Two new directors of **Micro Data Base Systems** (Lafayette, IN), **Donald P. Eckrich** and **William G. Bollinger**, come from very different fields, both quite distinct from the micro industry. Eckrich has been president of Beatrice Foods in Chicago, while Bollinger's background is in investment research, most recently with the New York firm of Goldman, Sachs. Recently, Micro Data Base Systems completed a private placement totalling about sixty thousand dollars.

□ Joining a growing tide of corporate moves to our potential fifty-first state, **Integral Data Systems** (Milford, NH) is opening a new manufacturing subsidiary in Puerto Rico. **Integral Data Systems de Puerto Rico** is being guided by subsidiary president **Donald M. Decker** and general manager **Alfredo M. Gonzalez**. The plant, located about twenty-five miles southeast of San Juan in Juncos, will employ seventy people by the end of 1981. Subsidiary is manufacturing printed circuit boards for company's Paper Tiger printer, resulting in complete inhouse control of all the printer's components for the first time.

□ **Microtek** (San Diego, CA) is creating a new division called **Microtek Peripherals**, with Microtek vice-president **Daniel Obed** at the helm. The division will function as a research and development unit for micro software, peripherals, and parallel and serial interface cards for the Apple II. Back at headquarters, the company has named **Diane Barney-Laukat** as vice-president of marketing. She will guide the planning of a new line of low-cost matrix printers geared to compete with the recent influx of matrix printers from abroad.

□ A national search for the most innovative and creative applications of the personal computer for handicapped individuals is currently being conducted by an organization at Johns Hopkins University, **Personal Computing To Aid the Handicapped**. Projects designed to use the Apple are welcome. Funded by grants from the National Science Foundation and Radio Shack, the competition offers one hundred awards including a $10,000 grand prize, fifteen complete microcomputer systems, and numerous $1,000 prizes. In the interest of fairness, judging of creative applications will be conducted on three separate levels—professional, amateur, and full-time student. All rights to and future profits from the projects remain the sole property of the designer. Contest submissions are being accepted in three applications categories—computer-based devices, computer programs, and system concept and design. Applications for the entry information kit are available from Personal Computing To Aid the Handicapped, Johns Hopkins University, P.O. Box 670, Laurel, MD 20810.

□ The growth of **Apple Computer Inc.** continues with the company's recent purchase of **Microsense Computer Ltd.** in London, England. Microsense has been the authorized distributor of Apples in the United Kingdom since July 1979. Its former chairman, **Michael Brewer**, is now the subsidiary's managing director, while operations director **David Collis** and marketing director **Stephen Brewer** will remain in their current capacities . . . Apple's financial report for the first quarter of fiscal year 1981 shows sales 246 percent higher than in the same quarter in 1980 and net income nearly double that of last year at this time. **Michael M. Scott,** president and chief executive officer, has noted that "management considers it unlikely that we will sustain this rate of sales growth, on a consecutive quarterly basis, in the second fiscal quarter." The unlikely, it seems, happened. Early second fiscal quarter sales leapt far above forecasts, resulting in a monthlong, nationwide scarcity of Apples.

□ The computer service company, **Ross Systems** (Palo Alto, CA), has opened an office in Dallas, Texas, thus expanding its southwest United States market. Regional director in the Dallas office is **Cliff Hall**, a veteran in computer and timesharing industries. Ross Systems specializes in consultation and timesharing services and interactive software programs for managers in medium and large companies.

THE PASCAL PATH

By Jim Merritt

PASCAL Survival Guide

We concluded last time with a listing of our first program, "SomeExpressions":

```
PROGRAM SomeExpressions;
BEGIN
  WriteLn(2+2);
  WriteLn;
  WriteLn(2-2);
  WriteLn;
  WriteLn(2*2);
END.
```

The program merely calculates and displays the values of three arithmetic expressions. Each value is obtained by combining 2 with itself and a different arithmetic operator. These are addition (+), subtraction (−), and multiplication (*). The asterisk (*), rather than more conventional notation, was chosen for expressing multiplication to simplify the task of writing a compiler that understands and translates Pascal programs into object code. Many languages, including Basic and Fortran, use the asterisk as the multiplication operator for the same reason.

Getting Down to Business. Although some computer programs are written only to be read and discussed, most, including "SomeExpressions," are written to be executed. To do this, you must put the program into the computer and compile it; so, we must now postpone our look at the Pascal language in favor of learning to use the three Apple Pascal system components: the operating system, the editor, and the compiler.

This survival guide is an abbreviated discussion of the process necessary to enter, compile, and execute most example programs you will encounter, including "SomeExpressions." It deals only with the essential concepts you must learn and the actions you must perform to get the job done and to understand a little about what you've done once you are finished.

The summary is terse, but, if you do everything as it's given without deviating or making a serious mistake, all should happen as described. Should you encounter difficulties or want to know more about the system, please read carefully the relevant sections in Apple's two well-written Pascal manuals.

The Operating System. When you boot Apple Pascal, the operating system is executed automatically. It gives you a welcome message, displays its *main prompt line,* then waits for you to press a key corresponding to one of its single-character commands. To enter, compile, and execute "SomeExpressions," you'll use three of these commands:

1. *E* causes execution of the program named SYSTEM.EDITOR—the screen editor that you use to enter "SomeExpressions" into the computer and save it on diskette.

2. *C* causes execution of SYSTEM.COMPILER. The compiler will generate an object code program corresponding to the Pascal source program you specify, in this case, "SomeExpressions."

3. *X* causes execution of any object program you specify.

For now, you may think of the E and C commands as specialized, restricted versions of X, in that they result in the execution of specific programs (SYSTEM.EDITOR and SYSTEM.COMPILER, respectively), while X permits the execution of any arbitrary program.

As soon as you press an alphabetic key corresponding to a legal command, the operating system attempts to execute that command. That is, if you press the E key, the system begins searching all mounted diskettes for SYSTEM.EDITOR, and executes that program if it can be found. The system does not wait for you to press any other key; E is enough.

One-Keystroke Commands: Good and Bad. Apple Pascal's orientation toward single-keystroke commands is aimed at reducing your typing burden. The system was designed to be as friendly as possible to the hunt-and-peck typist. Unfortunately, when, sooner or later, we accidentally press a wrong key, the system executes whatever legal command was specified by the mistaken keystroke. This can be frustrating, but it is very rarely disastrous.

Recognizing this problem, the system designers provided many opportunities for you to back out gracefully when you make a mistake. We'll examine each of these opportunities as it becomes relevant. For now, be careful while typing—taking things slowly and deliberately is never a bad way to start out.

Rough Road Ahead for One-Drive Systems. You'll find Pascal rough going if you have only one disk drive in your system. Smooth program development is possible in the Pascal system only when all of the tools you need, the operating system, editor, and compiler, are all available at once, without your having to switch diskettes. The major components of the Pascal system are distributed between two diskettes, APPLE1: and APPLE2:, because they won't all fit on a single floppy. Apple provides a way around this by supplying the APPLE0: diskette, for use with single-drive systems.

You can read about how APPLE0: is used in the "One-Drive Startup" section of the *Apple Pascal Language Reference Manual*. However, although APPLE0: makes limited program development possible on a single-drive system, it does not make it easy to do or to explain. Therefore, this column assumes that you have at least two drives, and that, throughout this discussion, the main drive contains a *copy* of APPLE1: and the second one contains a *copy* of APPLE2:. If you have only one drive, the information in "One-Drive Startup" should enable you to follow along reasonably well.

The Screen Editor.

Entering the Editor. To begin putting "SomeExpressions" into your Apple, press the E key, which starts the editor. The editor will display a prompt line informing you that there is no *workfile* and asking you to specify a file to edit. Shortly, we'll go into more detail about what a file is, but now we are more concerned with typing in the program.

At this point, press only the return key, indicating that you wish to begin this editing session with a clean slate for entering and revising completely new material. Here is one of the numerous backing out points: if, instead of responding by pressing only the return key, you press the ESC key, then the return key, the editor will cease execution, and you will receive the system's main prompt line once more. Throughout the system, the ESC key is often recognized as a sort of panic button, permitting you to undo what you've just done. In this case, you can undo your entrance to the editor.

Prompt Lines, Buffers, Cursors, and Commands. When you respond to the first prompt by pressing only the return key, the editor clears the screen then displays the *editor prompt line*. Below this line is an empty screen representing the contents of the editor's *main buffer*, which we'll call simply the buffer. The buffer is the place in the Apple's memory in which text is stored while you are entering or revising it.

Because you haven't yet entered any text, the buffer is empty. The exact spot where you are working is marked by the *cursor*, which always sits at the character position into which you are *about* to type. Since the buffer is now completely empty, the cursor is at the very first position of the *buffer-display window*—that is, just below the editor prompt line.

The editor responds to single-key commands just as the operating system does. These commands are summarized in the editor prompt line. To enter the source text for "SomeExpressions," you need only use *I* (*i*nsert), and possibly *D* (*d*elete) and *X* (e*x*change). Note that the X key, which means one thing (execute) to the operating system, means something quite different to the editor. Because different system components often recognize the same alphabetic keys as entirely different commands, you should get in the habit of checking the prompt at the top of the screen to determine not only which component you are using at any given time, but also its menu of permissible single-key commands.

Insert Mode. Press the I key to tell the editor that you wish to insert text. The prompt line will change to indicate that you are now in the *insert mode*. In this mode, all the visible characters you type will be taken as text, put into the buffer, and displayed on the screen. The editor does not respond to one-keystroke commands while it is in insert mode.

There are two ways to leave insert mode and regain the editor prompt line. One is to press the control key while *simultaneously* pressing the C key. The other is to press ESC. From now on, we will refer to the procedure of pressing and holding down the control key while pressing another character merely as control-N where *N* is the character.

The control-C signal has been given a proper name by computer scientists: *ETX*. On prompt lines and in Apple system documentation, you will see ETX and control-C used interchangeably. Both terms refer to the simultaneous two-key couplet of control and C. When the editor is in insert mode, control-C signals it to accept into the buffer all text you have entered up to that time. When you press control and C together while the editor is in insert mode, you are, in effect, saying, "I have no more new text to enter for now. Please accept what I have given you and let me use the one-keystroke commands once more."

Exiting this mode by pressing ESC is pushing the panic button. This signals the editor to return the buffer and corresponding screen window to the states they had before you entered insert mode. ESC *cancels* any insertion in progress, *throwing away* any text you entered while the editor was in insert mode. Of course, any text that existed in the buffer *before* insert mode was entered remains unaffected by the abortive action of ESC.

With the editor in insert mode, you are finally able to type "SomeExpressions" into the computer. Simply type the text given in the printed version of "SomeExpressions," just as you would with any typewriter. The text will appear on the screen as you type. As with a typewriter, you must indicate that you have finished the current line and wish to begin a new one by pressing the return key.

In this column, we distinguish between Pascal keywords and identifiers by printing program titles in mixed upper-case and lower-case and keywords in capital letters only. The compiler, however, makes no distinction between the two cases in translating programs. Using the standard Apple video display and keyboard, you will seem to be able to enter only upper-case information. Don't worry. The program you enter will compile into exactly the same object code as the nicely printed one. One thing the editor will help you to do, however, is preserve the *indentation* of the printed version.

Automatic Indentation. Pascal is one of many languages that permit arbitrary indentation of program lines. Indentation is a good way to partition the text of a program into recognizable chunks that can be taken in at a glance, and I will use it often in publishing programs and program fragments for you to study. During insert mode, you indent by pressing the space bar until you reach the desired position on the line. When you finish typing text on that line and press the return key to advance to the next display line, the editor *remembers* the amount of indentation you used, and indents the next line exactly that much. If you then indent even further, that, too, is carried over to succeeding lines.

To cancel some or all of the accumulated indentation, use the left-arrow key at the very beginning of a new program line. The cursor will move left one space at each depression of the left-arrow key, and the *indentation memory* will be decreased by one unit. Be careful not to press the left-arrow key when the cursor is in the farthest position to the left on the line or the cursor will end up on the previous line, and the indentation memory will be *reset* to whatever indentation that line has.

Correcting Errors in Insert Mode. If you're careful (and lucky), you'll be able to enter "SomeExpressions" into the computer without making a mistake. However, watch the screen to see what you type. If you do make an error and discover it *before* you press the control-C couplet to end insert mode, you may use one or both of two special keys to correct the problem.

1. *Single-character erasure* or *backspacing.* The left-arrow key works like a correcting backspace key on a sophisticated electric typewriter. It erases the character you typed last during this session in insert mode and permits you to type another character into the now-vacant position. You may erase several characters in a row, such as a word or phrase, by pressing the left-arrow key more than once. (Note that, when you are changing the indentation memory by pressing the left-arrow key, you are merely erasing spaces.

On an electric typewriter, you may backspace to correct only until you reach the left margin of the current line. Using the screen editor, you may backspace until you have erased the *first character entered* during the current insert mode session. In our example, the first character in the current session is the very first character in "SomeExpressions." If you are not averse to retyping this small program, you might try erasing it entirely, character by character. While you are doing this, notice how the cursor travels not only right-to-left on the line as you backspace, but also from line-to-line, bottom-to-top.

When you have backed over the very first character entered in the current insert mode session, press the left-arrow key once more. You should get an error message at the top of the screen, indicating that you have tried to backspace too far. As the message indicates, you must press the space bar to eliminate the error message and continue in insert mode.

2. *Line erasure with control-X.* Perhaps erasing on a character-by-character basis is not fast enough or convenient for you. You may erase the entire line you are working on by pressing control-X. When the line has been erased, the cursor will be positioned just to the right of the last character on the previous text line.

As with single-character erase, you may repeat control-X as many times as you wish. Unlike the backspace, however, control-X cannot erase *all* the text entered during the current insert mode session. You can repeat control-X until the cursor is just to the right of the last character on the first line in the program. If you press control-X again, the editor will try to erase the line and position the cursor on the right side of a previous line that does not exist. Because it can't do this, it refuses to erase the one remaining line and gives you the same error message at the top of the screen as it did when you tried to backspace too far with the left-arrow key.

Review of Insert Mode. Insert mode permits you to type information into your computer, using it as a sort of electric typewriter. You may enter insert mode by pressing the I key when you see the editor prompt line at the top of the screen and may leave it by pressing control-C. Ending insert mode in this way fixes the text you have entered permanently within the editor buffer and screen display window. To abort insert mode and throw away any text you have entered during the current insert mode session, press ESC. To let the editor know when you are finished typing a line of text, press return. The editor will remember and propagate from line to line any indentation you use.

The left-arrow key is used as an erasing backspace for cancelling single-character mistakes and may be repeated to erase several characters in sequence. Control-X cancels an entire line at a time and may be repeated to erase several lines in sequence except the first line of the current insert mode session. Neither the left-arrow key nor control-X may be used to erase more text than you have entered in the current insert mode session.

Cursor Movement. When you've returned to the editor prompt line, press the left-arrow key again. *No erasure occurs*; rather, the cursor moves one character closer to the start of the text. The editor interprets the left-arrow key as a cursor-positioning signal whenever you can see the editor prompt line. Try the right-arrow key, which has no significance in insert mode. It now moves the cursor toward the end of the text. Pressing the space bar gives you the same result at the right-arrow key. Pressing return repositions the cursor at the beginning of the next display line on the screen. None of these movements affects the text. These are examples of cursor movement signals you can give the editor only when you see the editor prompt line.

Sometimes you'll wish to move up through the text more quickly than the left-arrow key will permit or down through the text without changing the cursor position to the beginning of each line. To move the cursor one line *higher* on the screen at a time, press control-O. This may be considered an up-arrow signal. Similarly, use control-L to drop the cursor one line lower—a down-arrow signal. Note the difference between the action of return and control-L. Return always puts the cursor at the beginning of the next line; control-L drops the cursor to the next line, retaining its horizontal position.

Play with the cursor movement signals until you are comfortable with the way they work. Note that your boundaries are the start and end of the text buffer. You can't regress past the first character in the buffer, nor can you advance beyond the last. Leave the cursor somewhere in the middle of your text, ready for experimenting with the delete and exchange modes.

Delete Mode. Entering delete mode involves pressing the D key when you can see the editor prompt line. In delete mode, you may eliminate text from the buffer and the screen. Deletions begin at the current cursor position. The cursor movement signals control the direction of deletion, right, left, up, or down. With the editor in delete mode, try giving some of the cursor movement signals.

Notice that if you change directions during deletion the deleted text is *restored*. As an example, let's say that you can see the editor prompt line and decide to delete the keyword *BEGIN*. First, move the cursor until it is superimposed over the *B*. Then enter delete mode by pressing the D key. Finally, erase *BEGIN* by pressing the right-arrow key five times.

Having erased *BEGIN*, press the left-arrow key a couple of times and you'll get back that many letters of the deleted word. Any deletion you make does not actually affect the text buffer until you leave delete mode via control-C.

As with the insert mode, control-C is a signal to the editor that you are satisfied with your work and wish to make it permanent. Until you give this signal, you may abort the deletion by pressing the ESC key or move the cursor in the opposite direction to recover deleted material. If you use the panic button, the editor will instantly update the screen window display to reflect the state of the buffer before your aborted attempt at deletion. If you end the delete mode with control-C, the remaining text on the screen will collapse together.

Exchange Mode. This mode permits you to overwrite existing text with new, presumably different, characters. To enter it, you press the X key when you can see the editor prompt line. In this mode, you may overwrite the characters from the current cursor position through the end of the line. The editor will not allow you to skip to previous or succeeding lines nor to add any new text past the last character position on the line. If the cursor is already at the end of a line when you enter exchange mode, you won't be able to do *anything* in it.

To see how exchange works, assume the cursor is superimposed over the B on a line containing only the keyword *BEGIN*. If you enter exchange mode, you will be able to type over *BEGIN*, replacing it with *ABCDE*—but *not* with *ABCDEF*, because there are only five character positions to be overwritten on that line.

An exchange, like a deletion, doesn't become permanent until you end exchange mode by pressing control-C. You may abort the exchange by pressing ESC. Only one of the cursor movement signals, the left-arrow backspace, is obeyed by the editor when it is in exchange mode. Backspacing in exchange mode recovers the original text character by character.

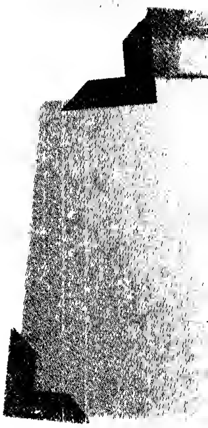
Shaping Up Your Program. Experiment with the three modes and the cursor movement signals, keeping in mind that deletions and exchanges begin *at* the current cursor position, but that inserted material will be injected *immediately in front of* the current position. Look for discrepancies between your version of "SomeExpressions" and the one printed in this column. If there are any differences (other than the use of only upper-case letters), make use of the cursor movement signals and the three operating modes to fix them. When your on-screen copy of "SomeExpressions" corresponds exactly with the printed one, congratulations!

Writing the Program to Diskette. Make sure to leave insert mode, delete mode, or exchange mode by pressing control-C. It is time to write "SomeExpressions" onto diskette. You should now be able to see the editor prompt line. Press the Q (quit) key. You will be shown several options, any of which you may choose by pressing the corresponding key.

Since this is the first time the program will be written to diskette, press the W key, specifying the write option. (Do *not* press the E key, which corresponds to exit; this option is used when it is not necessary or desirable to save the results of an editing session.) The write option permits you to specify the

The Newest Face at Summer Camp Is Apple's

BY KIRIN TOMMERVIK WITH PHOTOGRAPHS BY KURT A. WAHLNER



Computer Camp is a fun place to go to play games and to learn how to program. You work with nice people such as Denison Bollay, manager; Garry, Basic instructor; Sue, Basic instructor; Christopher Wells, Pascal instructor; Peggy, cook and nurse; Mark, counselor, and Susan Reid, cook. I must congratulate them for a wonderful job of making a Computer Camp.

While you have fun at Computer Camp, you also learn more about computers in general besides learning how to program graphics and other stuff. I would suggest Computer Camp to any youngsters who want to learn how to program. Computer Camp is a very fun place to relax and to play with computers.

Some of the games at Computer Camp are: tennis, volleyball, capture the flag, swimming, and horseback riding. ▪

The computer room with a couple of kids playing with some computers.

Softalk Presents The Bestsellers

The most notable event in the month of February, the results of which are charted here, was the resurgence of serious applications software in an otherwise lazy software market.

No new business or home/hobby programs penetrated the Top Thirty, but the level just below that—running approximately from forty-fifth to thirty-first—is filled with serious applications software. Sales were so close that the six programs not included in the Top Thirty that did make the Business Ten list are all within a whisker of thirtieth position.

It is not clear whether the trend will continue or whether February applications sales reflect pent-up demand caused by a combination of shortages in January. Many of the sales delivered in February were actually made in January, but delivery was delayed pending hardware or software.

Bucking the business trend, *Apple Galaxian*—soon to be known as *Alien Rain*—managed to stave off *VisiCalc* to remain the top program for the third consecutive month. The narrowing margin of difference portends a *VisiCalc* victory in March, but *Galaxian*'s lead over the next closest competitor would seem to indicate the likelihood that it would stay at the top of the recreational list.

Business 10

This Month	Last Month	
1.	1.	*VisiCalc,* Software Arts/Dan Bricklin and Robert Frankston, Personal Software
2.	2.	*Apple Plot,* Apple Computer
3.	4.	*DB Master,* Alpine Software/Stanley Crane and Barney Stone, Stoneware
4.	3.	*Data Factory,* Bill Passauer, Micro Lab
5.	5.	*Apple PIE,* Tom Crossley, Programma
6.	6.	*Apple Writer,* Apple Computer
7.	—	*Information Master,* James A. Cox and Stephen M. Williams, High Technology
8.	—	*Personal Filing System,* John Page, Software Publishing Company
9.	10.	*BPI General Ledger,* John Moss and Ken Debower, BPI
10.	9.	*Easy Writer,* John Draper, Information Unlimited Software

Early indications in March are that *Space Eggs* from Sirius Software and *Zork* from Personal Software may be *Galaxian*'s most serious competitors for that month.

Most of the dynamic movement and interesting battles were taking place among serious applications software during February. *DB Master* overtook *Data Factory* to become the bestselling data base program. The battle continues to heat up as Micro Lab introduces a more powerful version of *Data Factory* as well as their *Mini Factory*.

Equally noteworthy was the presence of two new data base programs in the Business Ten. *Personal Filing System* from Software Publishing and *Information Master* from High Technology now appear ready to compete on equal terms.

The word processing battle continues unabated as well. For the second month, Programma's *Apple PIE* leads the genre, but the margin over *Apple Writer* and *Easy Writer* is statistically insignificant and *SuperText* and *Magic Window* are but a few steps behind.

Softalk Presents The Bestsellers

BPI's *General Ledger* package continues to dominate the accounting software.

Making perhaps the biggest impact in business software is a piece of hardware—the *SoftCard* from Microsoft. With several thousand of them now in the hands of business users, CP/M-based software is starting to find acceptance in the Apple market. The *Magic Wand* word processor and Peachtree's accounting packages are the first of what may become a new flood of competitors.

For some reason, February was the month for stock market investors, as reflected in the Home/Hobby Ten. *Market Charter* from RTR Software and Apple Computer's *Dow Jones Portfolio Evaluator* made their first appearances in that list. In addition, *Dow Jones News and Quotes Reporter* from Apple Computer made the list for the second time.

The only three programs to crack the Top Thirty were *Akalabeth, Star Warrior*, and *Reversal. Akalabeth* made the list by virtue of reasonably steady sales while other programs were dropping; except during the Christmas season, when it made twenty-third (January 1981), it has held between thirtieth and fortieth. *Star Warrior* and *Reversal* are relatively new offerings that may be expected to improve their showings in future months.

Making the largest jump in the Top Thirty was *DB Master*, moving from twenty-seventh to twelfth. Also making sizeable gains were *Apple Plot*, going from thirteenth to fourth, and *Typing Tutor*, which gained from fifteenth to ninth.

The renaissance of *Typing Tutor* is attributed to the release of the Applesoft version on disk.

Remaining as stolid performers of great consistency were the old reliables. In this category fall *Flight Simulator* from SubLogic, *Sargon II* from Hayden, *Super Invader*, from Creative Computing, and *Adventure*, from Microsoft.

Bidding fair to join that group as an all-time bestseller of great consistency is *Temple of Apshai*, Automated Simulations's first role-playing fantasy, which continues to perform well even as its subsequent follow-ups trail off.

Ken Williams of On-Line Systems remains the hottest programmer extant, with four collaborations in the Top Thirty. Nasir of Sirius has three offerings at that level and Bill Budge of California Pacific has two. Dan and Kathe Spracklen join the list of programmers with multiple programs in the Top Thirty with *Reversal* accompanying *Sargon II*.

Home/Hobby 10

This Month	Last Month	
1.	2.	*Typing Tutor*, Image Producers, Microsoft
2.	1.	*DOS 3.3*, Apple Computer
3.	3.	*DOS Tool Kit*, Apple Computer
4.	4.	*Bill Budge's 3-D Graphics Package*, Bill Budge, California Pacific
5.	6.	*LISA Assembler*, Randy Hyde, Programma
6.	—	*Market Charter*, B. C. Burch, RTR Software
7.	—	*Dow Jones Portfolio Evaluator*, Apple Computer
8.	—	*Higher Text*, Ron and Darrell Aldrich, Synergistic Software
9.	—	*Dow Jones News and Quotes Reporter*, Apple Computer
10.	7.	*Enhanced Paper Tiger Graphics*, David K. Hudson, Computer Station

Apple-franchised retail stores representing approximately 6 percent of all sales of Apple and Apple-related products volunteered to participate in the poll.

Respondents were contacted early in March to ascertain their sales leaders for the month of February.

The only criterion for inclusion on the list was number of sales made—such other criteria as quality of product, profitability to the computer retailer, and personal preference of the individual respondents were not considered.

Respondents in March represented every geographical area of the continental United States as well as Hawaii and Alaska.

Results of the responses were tabulated using a formula that resulted in the index number to the left of the program name in the Top Thirty listing. The index number is an arbitrary measure of relative strength of the programs listed. Index numbers are correlative only for the month in which they are printed; readers cannot assume that an index rating of 50 in one month represents equivalent sales to an index number of 50 in another month.

Probability of statistical error is plus-or-minus 7.2 percent, which translates roughly into the theoretical possibility of a change of five points, plus or minus, in any index number.

The Top Thirty

This Month	Last Month	Index	
1.	1.	91.81	*Apple Galaxian*, Tony Suzuki, Broderbund Software
2.	2.	88.86	*VisiCalc*, Software Arts/Don Bricklin and Robert Frankston, Personal Software
3.	5.	47.56	*Hi-Res Adventure #2: The Wizard and the Princess*, Roberta and Ken Williams, On-Line Systems
4.	13.	37.24	*Apple Plot*, Apple Computer
5.	4.	35.03	*ABM*, Silas Warner, MUSE Software
6.	3.	34.66	*Flight Simulator*, Bruce Artwick, SubLogic
	9.	34.66	*Phantoms 5*, Nasir, Sirius Software
8.	6.	33.93	*Hi-Res Adventure #0: Mission: Asteroid*, Roberta and Ken Williams, On-Line Systems
9.	15.	31.71	*Typing Tutor*, Image Producers, Microsoft
10.	7.	28.02	*Hi-Res Football*, Jay Sullivan and Ken Williams, On-Line Systems
11.	8.	26.92	*DOS 3.3*, Apple Computer
12.	27.	24.70	*DB Master*, Alpine Software/Stanley Crane and Barney Stone, Stoneware
13.	20.	23.23	*Super Invader*, M. Hata, Creative Computing
14.	11.	21.01	*Hi-Res Adventure #1: Mystery House*, Ken and Roberta Williams, On-Line Systems
15.	19.	20.65	*DOS Tool Kit*, Apple Computer
	12.	20.65	*Adventure 10: Savage Island*, Scott Adams, Adventure International
17.	17.	20.28	*Hellfire Warrior*, Automated Simulations
18.	14.	19.91	*Sargon II*, Dan and Kathe Spracklen, Hayden
19.	18.	19.17	*Cyber Strike*, Nasir, Sirius Software
20.	29.	18.80	*Adventure*, Software Associates, Gordon Letwin, Microsoft
21.	10.	18.07	*Dogfight*, Bill Basham, Micro Lab
22.	29.	15.49	*Bill Budge's 3-D Graphics Package*, Bill Budge, California Pacific
23.	26.	15.12	*Data Factory*, Bill Passauer, Micro Lab
	23.	15.12	*Temple of Apshai*, Automated Simulations
25.	28.	14.01	*Star Cruiser*, Nasir, Sirius Software
26.	—	12.90	*Akalabeth*, Lord British, California Pacific
27.	15.	12.54	*Odyssey*, Bob Clardy, Synergistic Software
28.	20.	12.17	*Bill Budge's Space Album*, Bill Budge, California Pacific
	—	12.17	*Star Warrior*, Automated Simulations
30.	—	11.80	*Reversal*, Dan and Kathe Spracklen, Hayden